Embedding STEAM in Early Childhood Education and Care

Edited by
Caroline Cohrssen · Susanne Garvis

palgrave
macmillan

Editors
Caroline Cohrssen
Faculty of Education
The University of Hong Kong
Hong Kong SAR, China

Susanne Garvis
Department of Education
Swinburne University of Technology
Hawthorn, Australia

ISBN 978-3-030-65623-2 ISBN 978-3-030-65624-9 (eBook)
https://doi.org/10.1007/978-3-030-65624-9

This Palgrave Macmillan imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

5

Screen-Free STEAM: Low-Cost and Hands-on Approaches to Teaching Coding and Engineering to Young Children

Amanda Sullivan and Amanda Strawhacker

Introduction

In the United States and worldwide, there has been a growing focus on promoting Science, Technology, Engineering, and Mathematics (STEM) education during the early childhood and elementary years (National Science and Technology Council, 2018). This may be due in part to the noticeable lack of professionals qualified to take on jobs in the sciences. In less than a decade from now, it is estimated that the United States will need 1.7 million more engineers and computing professionals (Corbett & Hill, 2015). Early childhood and early elementary school is a critical time to reach future scientists and engineers in order to meet this growing workforce need (Bers, 2012, 2018; Sullivan, 2019). Children who are exposed to STEM curricula and programming at an early age

A. Sullivan (✉) • A. Strawhacker
Tufts University, Medford, MA, USA
e-mail: amanda.sullivan@tufts.edu

C. Cohrssen, S. Garvis (eds.), *Embedding STEAM in Early Childhood Education and Care*, https://doi.org/10.1007/978-3-030-65624-9_5

demonstrate fewer gender-based stereotypes regarding STEM careers, an increased interest in engineering, and fewer obstacles entering these fields later in life (Madill et al., 2007; Markert, 1996; Metz, 2007; Steele, 1997; Sullivan, 2019; Sullivan & Bers, 2017). Moreover, we have seen many cognitive and social benefits of implementing STEM, and particularly computer science, robotics, and engineering curricula with young children (e.g. Bers, 2008; Fessakis, Gouli, & Mavroudi, 2013; Kazakoff, Sullivan, & Bers, 2013; Lee, Sullivan, & Bers, 2013).

Despite the research, actually *reaching* children with quality STEM content, particularly with regard to the "T" of technology and "E" of engineering, during their foundational early childhood years has proven to be a real challenge to many parents and educators. Choosing developmentally appropriate ways to address fields like engineering and computer science with young children presents both practical and ethical issues for adults to consider. Innovative technologies to support STEM learning such as iPads, robotics kits, and computers are expensive, and often the cost of these materials (let alone the cost of training and professional development for adults on how to use them effectively) makes them out of reach for many parents and educators. This has opened the door to a new type of "digital divide" in which some schools and homes have access to high-quality STEM and computing devices while others do not.

In this chapter, we present a different approach to exploring technology, engineering, and the sciences during the early childhood years. By focusing on screen-free, low-tech, and collaborative approaches to topics such as engineering and coding, we demonstrate that it is possible to teach and learn technical STEM skills without access to expensive digital technology and kits. These inclusive activities are designed to be accessible to children of any gender and background within the age range of approximately three to eight years of age, and can be implemented in both home and school settings. The activity examples within this chapter highlight a Science, Technology, Engineering, Arts, and Mathematics (STEAM) rather than STEM approach to designing curriculum. By integrating the arts with the sciences, this chapter explores the ways that domains such as computer programming and engineering can be enhanced by infusing opportunities for creativity and artistic expression as well as an integration with other early childhood curricular content.

STEAM in Early Childhood Education

The STEAM Movement

Early childhood STEM education has historically focused on building foundational numeracy skills and an understanding of the natural sciences for young children (Bers, 2008; Bers, Seddighin, & Sullivan, 2013; Moomaw & Davis, 2010). In the growing national and international level discussion around STEM, *how* to effectively teach technology and engineering has become more pressing to researchers and educators (National Science and Technology Council [US], 2018; UK Department of Education, 2013; US Department of Education, 2010). This concept of promoting creativity and expression through technology and science is articulated in a newer acronym called "STEAM" (Science, Technology, Engineering, Arts, Mathematics) that is growing in popularity across the United States and worldwide (Allen-Handy, Ifill, Schaar, Rogers, & Woodard, 2020; Watson, 2020; Yakman, 2008). The "A" of STEAM can represent more than just the visual arts, but also the liberal arts, language arts, social studies, music, and more.

Within an early childhood context, STEAM education means finding ways for children to explore these subjects in an integrated way through hands-on projects, books, discussions, experiments, art explorations, collaboration, games, physical play, and more. New technological tools such as programmable robotics kits and programming languages designed for young children have become a popular way to teach interdisciplinary STEAM content by integrating arts and crafts, literacy, music, and more with engineering and robotics (Barnes, FakhrHosseini, Vasey, Park, & Jeon, 2020; Bravo Sánchez, González Correal, & Guerrero, 2017; Elkin, Sullivan, & Bers, 2016; Sullivan, Strawhacker, & Bers, 2017). Robotics kits have evolved in the tradition of educational manipulatives that allow children to explore their understanding of shape and number, spatial relations, and proportion (Brosterman, 1997; Kuh, 2014; Nicholson, 1972; Resnick et al., 1998).

In research trials with simple robotics and programming languages, children as young as four years old have demonstrated understanding of foundational engineering, programming, and robotics content (Bers, Ponte, Juelich, Viera, & Schenker, 2002; Cejka, Rogers, & Portsmore, 2006; Sullivan, Kazakoff, & Bers, 2013; Sullivan & Bers, 2015; Perlman, 1976; Wyeth, 2008; Zviel-Girshin, Luria, & Shaham, 2020). In addition to mastering this new content, programming interventions have been shown to have positive benefits for children's developing numeracy, literacy, and visual memory, and can also prompt collaboration and teamwork (Clements, 1999; Lee et al., 2013). Moreover, we have seen young children use robotics kits to explore more than engineering and coding, including culture, dance, music, and more within an integrative STEAM context (e.g. Kim & Kim, 2020; Sullivan & Bers, 2017).

While there is growing evidence that programming education supports children's attitudes and interest in STEM fields, research is ongoing about the cognitive benefits of learning to code (Rodriguez, Rader, & Camp, 2016). Critics argue that it is unclear whether or how the knowledge that learners acquire when programming (often called computational thinking) can transfer to contexts beyond the coding environment (e.g. Greiff et al., 2014; Scherer, 2016). In a recent meta-analysis of transfer in computer programming education, the authors found a moderate overall transfer effect between computer science learning and other cognitive skills such as creativity, reasoning, mathematics, and metacognition (Scherer, Siddiq, & Sánchez Viveros, 2019). One conclusion from this work is that learners show high ability to apply programming knowledge in similar contexts to their learning environment, such as completing a novel task using a familiar programming platform (Scherer et al., 2019). This finding has yet to be confirmed in non-technological contexts, such as when children engage in "unplugged" (non-technological) coding activities (Hickmott, Prieto-Rodriguez, & Holmes, 2018). However, preliminary studies of the comparative effect of "unplugged" and technology-based coding activities on computational thinking found no differences between children who completed unplugged and those who used tablet-based coding activities (Messer, Thomas, Holliman, & Kucirkova, 2018; Rodriguez et al., 2016). Indeed, one study found that children who completed unplugged coding activities showed significantly higher computational thinking than a non-coding control group (Brackmann et al., 2017).

Although research is ongoing as to the cognitive outcomes of programming and robotics knowledge to different settings, researchers do tend to agree that unplugged coding and engineering activities are a useful way for children and adults to meaningful and positively engage with novel STEAM domains (e.g. Bell & Vahrenhold, 2018). Research on early childhood STEM education confirms that parents and teachers are critical for supporting children's positive early STEM experiences, but that they need training and resources to effectively foster STEM learning (Bell & Vahrenhold, 2018; McClure et al., 2017; Strawhacker, Lee, & Bers, 2017). This poses a challenge since teachers may not have had professional STEM training, but studies show that professional development experiences that teachers who used unplugged, story-based, and physical STEAM activities, like the ones we present in this chapter, expressed confidence and willingness to integrate STEAM domains into their classroom settings (Bell & Vahrenhold, 2018; Curzon, McOwan, Plant, & Meagher, 2014; Sentance & Csizmadia, 2017; Smith et al., 2015).

STEAM and the New Digital Divide

There are now many digital tools, such as the robotics kits previously mentioned, available for young children to explore STEAM. But many of these new tools, despite their benefits, are inaccessible due to the cost, technical support, and professional development needed to implement them properly. For example, the KIBO Robotics Kit, developed by KinderLab Robotics for children aged four to seven years, offers a screen-free and hands-on kit that has decades of research highlighting its educational benefits (e.g. Sullivan & Bers, 2015; Sullivan, Bers, & Mihm, 2017). But with a cost of $220–$500+ per kit, it is unfortunately beyond the budget of many early childhood educators and parents. Similarly, the LEGO WeDo robotics kit for children seven years and older costs $221 per kit and requires the use of a tablet or other device for programming. Bee-Bot, one of the cheaper robots for young children, is still around $60 per robot, without any other accessories, and without allowing for the building and constructing components that KIBO and LEGO WeDo offer. Even free coding applications and games require schools or homes to have access to tablets and computers for each child for them to be used as intended. The costs of these devices alone are already prohibitive to many, without factoring in fees and time for training and professional development for adults to feel confident using these tools with young children.

The stark costs of new coding and engineering materials for young children have opened the door to a new type of digital divide. The term "digital divide" once simply referred to whether classrooms or homes had computers and Internet access. Now that most homes and schools have Internet connectivity basic hardware, this phrase has taken on a new meaning. There is now a socioeconomic division between those with access to high-quality, open-ended software and technology that promotes creative STEAM learning and those who do not. For example, access to computer science classes and clubs is generally lowest for students from lower-income households (Busteed & Sorenson, 2015). Inequitable access to computer science education could place these students at a disadvantage as computer technology continues to advance, especially as coding is thought of as "the new literacy" in this day and age (Bers, 2018).

Low-Cost and Screen-Free Materials and Activities

Digital technology, games, robotics kits, and more can be wonderful ways to explore STEAM at the early childhood level (see Sullivan, Strawhacker, & Bers, 2017 for ideas on using robotics within a STEAM context). In this chapter, we simply hope to demonstrate that expensive technology is not the *only* way to teach coding and engineering to young children. In an attempt to reach all young children, we focus on presenting STEAM activity ideas and materials that are low-cost and accessible to all, in order to help bridge the divide in access to computer science and engineering education. All the activities and approaches can be done in homes or classrooms that are strictly technology and screen-time free, or they can be used to supplement other curricula that use computers, tablets, programmable robotics kits, and more. We will begin by exploring STEAM materials for toddlers to explore engineering and progress to materials, resources, and approaches for teaching computer science and engineering to children in Kindergarten through second grade.

Exploring STEAM with Toddlers

A screen-free and hands-on approach to exploring STEAM may be especially useful for those parents and educators working with young children under the age of four years. The American Academy of Pediatrics recommends that preschool-aged children, between the ages of two and five, should have limited screen-time each day (American Academy of Pediatrics, 2018). Therefore, STEAM exploration for very young children should focus on providing them with multisensory, hands-on experiences that engage their senses and build on their natural curiosity.

It is typically the "T" in STEAM that gives adults pause when thinking about reaching very young children. It is important to remember that technology does not just have to mean expensive electronic devices, computers, and tablets. Rather, we can think of technology simply as any human-made tool that allows us to solve a problem or complete a task more easily. When it comes to very young children, some developmentally appropriate tools to explore may include child-safe scissors, tongs, eye-droppers, magnifying glasses, ramps, and more. Toddlers can explore engineering and mathematics through building and experimenting with blocks, puzzles, building bricks, magnetic tiles, and more. Asking children questions and encouraging them to make hypotheses and observations while they play can help foster scientific inquiry and an engineering mindset.

Activity Example

In this example, a classic building activity from early childhood is augmented with prompts developed and tested by the authors during educational STEAM research interventions at schools, makerspaces, and weekend workshops with children. When exploring engineering and design with young children, building tall towers is one of the easiest activities parents and educators can implement (See Fig. 5.1). Young children naturally explore stacking with blocks and nesting cups when they are very young. By the time they reach preschool, many children are very adept at stacking (and knocking down!) structures. For this simple activity, almost any materials you have available can work, from blocks and building bricks to recycled materials like plastic cups, paper towel rolls, and more.

Fig. 5.1 Toddler-created tower built with magnetic tiles

This is an easy "free play" activity for children to explore on their own. Adults should focus on asking prompting questions that focus children on engineering and scientific method concepts. For example, adults could take this activity deeper through one or more of the following:

- Provide children with a variety of different tower-making materials. Encourage children to predict, or guess, which material will allow them to build the tallest tower.
- Ask children to think about what makes their tower sturdy, or strong. Is it having a wider base? Is it using a particular material?
- Help children to measure their towers and record the measurements.
- Encourage children to test and improve their towers. What is one thing children could change about their designs to make them more functional (i.e. stronger, taller, wider, etc.)?

While this activity has an explicit focus on science, mathematics, and engineering, it can easily integrate into a longer STEAM curriculum unit as well. For example, children can focus on the art and design of their towers by allowing them time to work with paints, crayons, or other craft materials that allow them to decorate and express their creativity. Children could also move on from building towers to building replicas of their own neighborhoods including houses, schools, supermarkets, and other neighborhood landmarks. This could be one part of a larger interdisciplinary social studies unit that focuses on community and mapping, but also on art and engineering, as children create and decorate community maps for their structures to sit upon.

Computer Science Unplugged for Young Children

For toddlers, we have seen that the "T" of technology in STEAM can focus on simple human-made tools like pencils, scissors, and more. The previous section focused on fostering engineering within an interdisciplinary STEAM context, rather than on technical areas like computer science. As young children grow older, they become more curious about how other elements in their human-made world around them work.

They wonder how things like cell phones and computers function. This becomes an opportunity to teach young children about technology and computer science. Children can learn that their favorite apps and digital games all work because of code. They can learn that just like they are learning to read and write in English, Spanish, or any other language, and they can also learn to read, write, and create code!

While there are many benefits to teaching coding to young children, the complications of screen-time and reliance on expensive devices present roadblocks in terms of accessibility. The "unplugged" approach to computer science education has become a powerful movement over the past two decades, as educators have recognized the value of integrating activities that do not require knowledge of computers or other technologies into the computer science curriculum (Bell & Vahrenhold, 2018). This unplugged (i.e. "tech free") approach focuses on teaching programming concepts through puzzles, games, art, and more, all without a computer, robot, or tablet. Unplugged approaches to computer science claim to enable the development of computational thinking without spending time or cognitive resources on syntax and grammar of programming languages (Bell, Alexander, Freeman, & Grimley, 2009; Bell, Witten, & Fellows, 1998). The original Computer Science Unplugged project was based at Canterbury University and has since been widely adopted internationally (translated into 12 languages), and it is also recommended in The Association of Computing Machinery (ACM) K-12 curriculum (Bell et al., 2009).

Computer Science Unplugged Activity Example

This example comes from free resources posted on the CS Unplugged website by the Computer Science Education Research Group at the University of Canterbury (the authors share no affiliation with this research group) (Bell et al., 2009). By visiting csunplugged.com, parents and educators can find a range of unplugged activities to implement with young children. The website has activities organized by topic and age range (see Fig. 5.2). For example, there is a list of activities to explore binary numbers, error detection, searching algorithms, and more. The

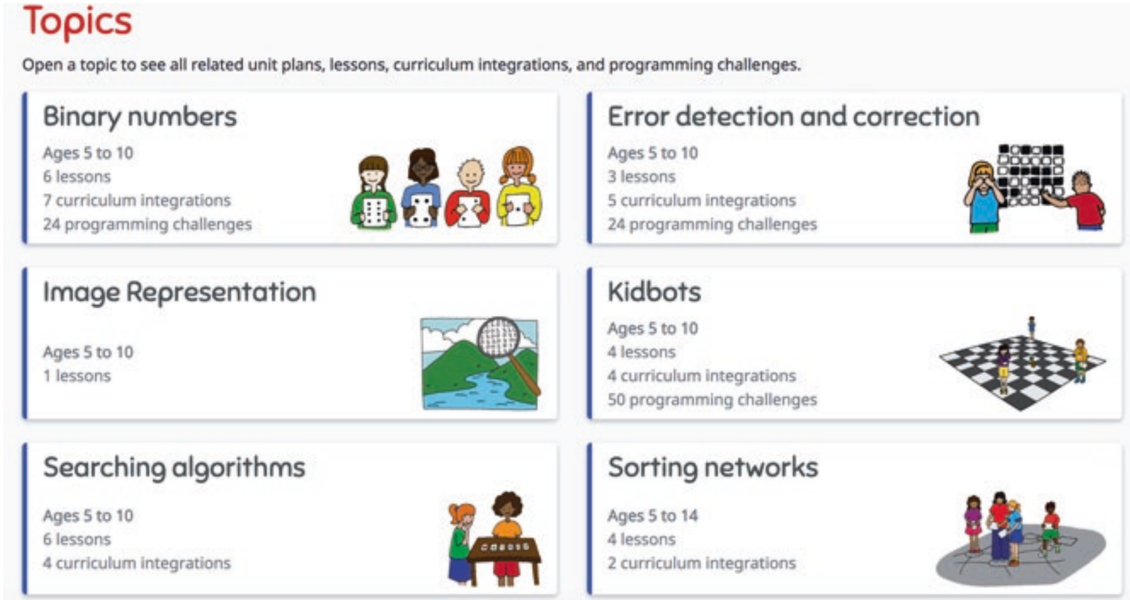

Fig. 5.2 Screenshot of activity topics on CS Unplugged website

prompts suggested for each topic typically involve a hands-on activity that may include the use of arts and crafts or other tangible materials, group discussion prompts, and ideas for play and exploration.

One lesson example presented for children as young as five years of age involves the binary number system. Why is the binary number system important for us to know about? Binary code is how computers talk and represent information. Children can think of binary as a fun new number language to explore. Children may be interested to learn that letters, numbers, and pictures (basically everything you see on the computer) is made up of different combinations of 0's and 1's.

Binary is a base-2 number system. This sounds complicated, but is just a bit different from the more common decimal, or base-10, number system. Every number "place" in our base-10 system is a multiple of 10, and we combine 10 digits (0–9) to create any number we want. For example, the number 158 only uses only three digits, but the *order* of the numbers matter: there is a 1 in the hundreds-place, a 5 in the tens-place, and an 8 in the ones-place. In binary, the system is exactly the same except that there are only 2 digits (0 and 1), and all the number places are multiples of 2. Computers use binary because it is simpler for a machine to understand than the complex decimal system. A 5-digit binary system can express numbers from 0–99,999. This is called "5 bit," and it actually is a shortened phrase that simply means "5 binary digits." CS Unplugged provides a helpful 5-bit binary to alphabet key that adults can adapt into posters or worksheets for their students (see Fig. 5.3). In 5-bit binary, each English letter can be represented by a combination of five 0s and 1s.

In the CS Unplugged activity, children create a necklace with their initials written in 5-bit binary. Adults do not need to cover the whole of binary to run this activity. Instead, this project is simply intended to be a fun and hands-on introduction to how computers store information. To complete this activity, decide which bead color will represent 1 and which bead will represent 0. For example, 0 could be blue and 1 could be red. Next, children choose their letters and see how their initials are translated into binary and then into colored beads. For example, the letter A (00001 in binary, see Fig. 5.3) would be represented by the following beads: blue, blue, blue, blue, red. To make this activity even simpler for young children, adults can create a poster showing a direct translation of

Base 10	Binary	Letter	Base 10	Binary	Letter
0	00000		14	01110	n
1	00001	a	15	01111	o
2	00010	b	16	10000	p
3	00011	c	17	10001	q
4	00100	d	18	10010	r
5	00101	e	19	10011	s
6	00110	f	20	10100	t
7	00111	g	21	10101	u
8	01000	h	22	10110	v
9	01001	i	23	10111	w
10	01010	j	24	11000	x
11	01011	k	25	11001	y
12	01100	l	26	11010	z
13	01101	m			

Fig. 5.3 Screenshot of alphabet to 5-bit binary key from CS Unplugged

how each letter of the alphabet would be represented by colored beads as well as worksheets for children to lay their beads on before stringing them.

In addition to exploring the concept of binary, activities like this can easily integrate into a longer STEAM curriculum that integrates art, fashion, and design in the creation of all sorts of jewelry, the creation of friendship bracelets, and more. While a deep understanding of binary is not needed for this activity, a basic understanding of what binary is can be helpful. CS Unplugged also provides a 45-minute lesson plan that can be used in conjunction with this activity that includes guided prompts on how to first introduce the topic of binary number system to young children, before getting into this hands-on activity.

Board Games to Explore Coding

From within the CS Unplugged movement, a new crop of unplugged coding board games and card games has been growing in popularity over the past decade. Board games and card games are some of the easiest (and most fun!) ways to explore computer programming with young children because you do not need a computer or any other device. Coding board games are also more conducive to learning and playing in home environments and informal education environments because they can be played with multiple players of mixed ages. Playing these games as a family can help younger children learn and understand the rules of the games faster than if they were to play by themselves. Table 5.1 outlines a few examples of popular coding board games that are designed to reach players younger than eight years of age. All of these examples are available for less than US$25, making board games a cost-effective solution for those without access to expensive tablets, computers, or robotics kits.

In addition to these coding-explicit games, parents and educators should remember that many traditional board games like Chess, Go and Backgammon can also be used to teach and reinforce the same problem-solving and strategy skills that are necessary across STEAM disciplines. Board games also help to teach young children important interpersonal skills such as patience, turn-taking, and being a gracious winner/loser.

Table 5.1 Coding board games for young children

Board game	Age range	Cost and description	STEAM skills
Robot Turtles	3+	\$21—multiplayer board game with the goal of programming your turtle to navigate a maze to reach its jewel.	Sequencing Problem-solving Debugging Functions Planning Turn-taking
LittleCodr	4+	\$13—a card game in which children program their parents or friends to do crazy things by using simple action cards	Logic Sequential thinking Prototyping debugging Turn-taking
Coding Farmers	7+	\$14—children play the game with action cards in two ways: regular English and Java code. By playing the game several times, children learn to connect their actions with written code.	Java programming Addition Subtraction Reading/vocabulary Turn-taking

Note. All prices given in USD

Board Game Example: Robot Turtles

This example comes from a board game developed by a software engineer who wanted a way to teach coding to his young children and was produced by a private company called ThinkFun (www.thinkfun.com) (the authors share no affiliation with the developers or producers of Robot Turtles) (Shapiro, 2014). The Robot Turtles board game teaches coding concepts to children ages three years and older, and was the most-backed board game in Kickstarter history at the time its campaign closed in 2014 (Shapiro, 2014). The game setup and rules of Robot Turtles are easy for young children to master: you create a maze on the board with the turtles in the corners and the jewels in the center (see Fig. 5.4). Children play instruction cards during their turn (such as, turn right, turn left, move forward, etc.) in order to "program" their turtles to get to their jewels. When a player's turtle reaches their jewel, they win. If they make a mistake, they can use a "Bug" card to undo a move. Creating these programs

Fig. 5.4 Robot Turtles board game

encourages sequential thinking and problem-solving: two key components of computer programming.

One of the great things about this particular board game is that the board can be set up differently each time you play, ensuring this is not a game you play once and then leave on a shelf in your closet. Additionally, it can increase in complexity as children grow up or become more familiar with the cards and logic of the game. For example, the "Function Frog" card is used to represent a set of several moves (i.e. it allows users to create a function or subroutine). By using this card, players learn to shorten their program by using this single card to represent a sequence of movements.

Robot Turtles can be a playful addition to family game night or used as a center activity in schools and informal education settings like camps or after-school programs. Parents and educators can take the board game concept further by encouraging children to design and create their own coding board games. This could develop into a longer STEAM curriculum unit that involves writing rules for the games, designing and decorating a playing board and/or cards, testing and improving the game by playing it with friends and peers, and more.

Books and Stories

Up to this point, we have focused on board games and hands-on materials or products to explore STEAM. But if parents and educators want to get started exploring STEAM in a very familiar way, it can be as easy as strategically integrating new books into your storytime practices. Adults can try reading and discussing engineering- or science-themed picture books and have discussions around what the characters did and why. A few notable examples include *Rosie Revere, Engineer* by Andrea Beaty; *Ada Twist, Scientist* by Andrea Beaty; *If I Built a Car* by Chris Van Dusen; *If I Built a House* by Chris Van Dusen; and *Going Places* by Paul Reynolds and Peter Reynolds.

Picture Book STEAM Activity Example

The authors developed this activity and reading list as part of their work, which included offering paid weekend-and-holiday STEM experiences for young children in the greater Boston area. Many picture books can naturally lead to a hands-on STEAM activity. For example, the book *If I Built a House* by Chris Van Dusen focuses on a child imagining the design of his dream house and all the fantastical elements it might include. After reading this book, children can create blueprints for their own dream houses, inspired by the blueprints in *If I Built a House*. Adults may also wish to facilitate a longer discussion about real-world structures, take a look at real building blueprints, and explore architecture from cultures around the world.

This can lead to a hands-on building activity that focuses on the engineering design process. The engineering design process refers to the cyclical or iterative process engineers use to design an artifact in order to meet a need. While there are many versions of the engineering design process, it typically includes a version of the following steps: identifying a problem, looking for ideas for solutions and choosing one, developing a prototype, testing, improving, and sharing solutions with others. Figure 5.5 shows one example of the engineering design process created by the Developmental Technologies Research Group at Tufts University.

After learning about the engineering design process, children can use a variety of materials such as LEGO, popsicle sticks, recycled materials, and more to bring their dream house blueprint designs to life (see Fig. 5.6). When faced with the actual materials at hand, some children may wish to revise their designs. All children should be encouraged to engage in the "test and improve" stage of the engineering design process by ensuring their houses are sturdy and implementing improvements or changes as needed.

From a STEAM perspective, this type of engineering activity can easily integrate more with fine arts, by incorporating a focus on painting, decorating, and considering the aesthetics of the houses. Or, it could integrate with literacy by connecting to a classic story such as *the Three Little Pigs*. Children could test the sturdiness of their houses against the breath of the "Big Bad Wolf" (i.e. a fan) and make any changes to their design based on the results of this test.

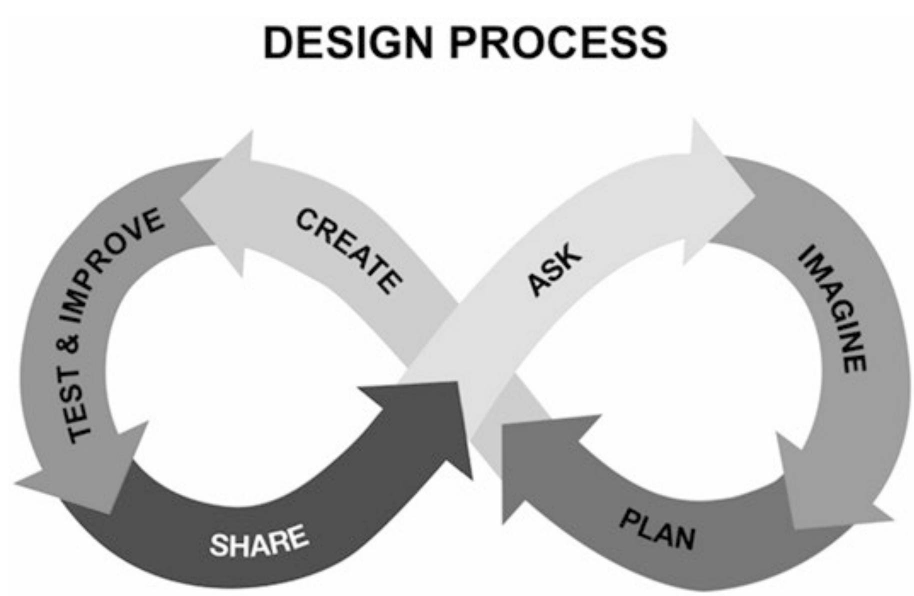

Fig. 5.5 The engineering design process

Fig. 5.6 Dream house creations made by children in K to Second Grade

Tips for Parents and Educators

Choosing appropriate tools and materials is only the beginning of what adults need to consider when it comes to implementing quality STEAM education for young children. Just as important as the tools we use, are the mindsets, attitudes, and role modeling to which we expose young children. This section focuses on providing parents and educators with tips, ideas, and resources for best practices exploring STEAM with young children.

Fostering a Growth Mindset

One of the most important things that parents and educators can do to support young children's STEAM education is fostering the right mindset toward hard work, perseverance, and failure. Research has shown that personal views about intelligence and failure may impact children's achievement and persistence in STEM fields. Psychologist Carol Dweck spent decades researching achievement and success and developed the concept of the "growth mindset" (Dweck, 2002, 2008). The "growth mindset" is the belief that intelligence is not fixed, but instead can change and grow incrementally through practice.

It is worth noting that Dweck's findings have met with criticism from the research community for her methodological approach, which her team attempted to address by launching a large-scale study of 12,000 students involving third-party research evaluators and methodological analysts. This confirmed core elements of her prior results, albeit with extremely small effects (Yeager et al., 2019). However, replication studies of Dweck's work, particularly randomized control trials, have met with mixed success. For example, Li and Bates (2017) found no achievement differences predicted by mindset in a sample of over 600 children, whereas Bettinger, Ludvigsen, Rege, Solli, and Yeager (2018) claim to have replicated Dweck's findings, and attribute their success to close adherence to Dweck's original intervention approach (see Denworth, 2019 for a full discussion of the ongoing debate about growth mindset). Despite this ongoing debate, education and psychology practitioners continue to use growth mindset in their practice, and researchers who support Dweck's work argue that educational interventions must be judged in a real-world context, where even small effects can be important (Denworth, 2019).

One way that adults can support a growth mindset is learning to praise children differently. Instead of simply telling children they are smart, which does not encourage growth, praise their *effort.* Praise the time and hard work children put into their project or mastering a new skill rather than just the outcome. Offering praise like "wow, you are so smart!" certainly can offer a short-term self-esteem boost, but in the long term, it can make children lose confidence when tasks become hard. Consider offering nuanced praise, such as, "I am so impressed that you spent so many hours working hard and building that LEGO house—I can tell it is really sturdy because of the wide base!" Not only does this type of praise help to foster the growth mindset, but it also shows you are paying close attention to their work, rather than offering generic compliments.

Role Modeling

Parents and educators should be aware of what and who young children are seeing on an everyday basis in school, at home, in the media, and in books. Are they exposed to engineers and scientists who look like them? Do they see women and minorities excelling at mathematics and using technology? Children need role models who reflect aspects of themselves that they can admire and look up to, especially within the sciences and technology. Adults can try to address this need by introducing young children to both fictional characters and real-life role models from STEAM fields that represent a range of genders and backgrounds. Some of the picture books referenced earlier in this chapter could be a great place to start. For example, *Rosie Revere, Engineer* and *Ada Twist, Scientist* both feature a female protagonist engaging in STEM.

Meeting real-world scientists and engineers can also be a powerful experience for young children. Educators can reach out to children's families for volunteers and may be surprised to find connections within your own classroom network. Parents and teachers can arrange trips to science museums, makerspaces, and laboratories for an exciting chance to meet or learn about scientists and engineers from a range of backgrounds. Local colleges and universities can also be a resource for finding diverse role models majoring in STEAM fields who may be interested in collaborating with you.

It is critical that parents and teachers do not forget about children's most impactful role models – the adults who care for them each day! Children are always watching and listening to what parents, teachers, and caregivers say and do. It is important for these adults to be modeling their own sense of scientific inquiry. How do you do this? You could start by pointing out to children when you have a hypothesis or idea that you are testing, demonstrate how you solved an engineering challenge, or share with children how mathematics or science knowledge helped you solve a problem in your everyday life. When you do not know the answer to a question a child asks, use this as an authentic opportunity to model problem-solving strategies rather than shying away from the question. In this way, you are modeling your own belief in the growth mindset and demonstrating your ability to apply the engineering and problem-solving skills you are teaching them.

Using the STEAM Resources Available

This chapter has focused on providing parents, educators, and caregivers with information about tools and approaches for teaching screen-free (and low-cost) STEAM activities to young children. But it is unlikely that anyone embarking on teaching early childhood STEAM for the first time would need to start from scratch. There are many resources available, both in-person and online. Within your own community, be sure to explore local children's libraries, museums, and makerspaces for STEAM-related resources and events. Settings like these will have access to tablets, computers, robotics kits, and other more expensive STEAM materials that you may be able to borrow or use without purchasing your own.

There are also many online resources from which parents and educators can benefit. From YouTube videos teaching the Engineering Design Process to free curriculum downloads, a variety of sites and resources support parents and early childhood educators on their STEAM journeys. For example, CS Unplugged, which was mentioned earlier in this chapter, has a range of activity and curriculum guides freely available online at csunplugged.com/. A few notable examples are as below:

- Code.org—Code.org offers many useful resources for parents and educators embarking on teaching computer science to children in grades K-12. As it relates to low-cost and screen-free activities, they have compiled a list of their unplugged curriculum ideas and resources here: code.org/curriculum/unplugged
- NASA for Educators—Lesson plans, teacher guides, classroom activities, posters and more for teachers and students as young as Kindergarten. www.nasa.gov/stem/foreducators/k-12/index.html
- Teach Engineering—A digital library comprised of standards-based engineering curricula for K-12 educators. See: www.teachengineering.org/

Conclusion

Young children are budding scientists and engineers who are naturally curious about the world around them and how things fit together and work. This means they are at a perfect age to explore STEAM and particularly, concepts of engineering and computer science. While there has been a growing focus on innovative new applications, digital devices, and software to encourage young children's exploration of computer science and engineering, there are also many low-cost and screen-free approaches to teaching the same concepts. Moreover, low-cost and low-tech materials and approaches may be useful in reaching schools and communities that are unable to afford new technologies and professional development for educators. Low-cost and low-tech STEAM approaches are accessible for parents and teachers, even those with little-to-no STEM background themselves. With this new crop of board games, card games, and unplugged activities, computer science and engineering is becoming more accessible to all.

Conflict-of-Interest Disclosure

The activities and resources presented in this chapter were developed by the authors during research-based and professional education interventions, or were available through third-party outlets (see specific examples for more information about their origins). The KIBO Robotics Kit described in the introduction is produced and marketed through KinderLab Robotics, Inc., where both authors have previously been employed. The authors are in no way affiliated with any of the other products described in the chapter.

References

Allen-Handy, A., Ifill, V., Schaar, R. Y., Rogers, M., & Woodard, M. (2020). Black girls STEAMing through dance: Inspiring STEAM literacies, STEAM identities, and positive self-concept. In *Challenges and opportunities for transforming from STEM to STEAM education* (pp. 198–219). Hershey, PA: IGI Global.

American Academy of Pediatrics. (2018). *Children and media tips from the American academy of pediatrics*.

Barnes, J., FakhrHosseini, S. M., Vasey, E., Park, C. H., & Jeon, M. (2020). Child-robot theater: Engaging elementary students in informal STEAM education using robots. *IEEE Pervasive Computing, 19*(1), 22–31.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology, 13*(1), 20–29.

Bell, T., & Vahrenhold, J. (2018). CS unplugged—How is it used, and does it work? In *Adventures between lower bounds and higher altitudes* (pp. 497–521). Cham, Switzerland: Springer.

Bell, T. C., Witten, I. H., & Fellows, M. (1998). *Computer Science unplugged: Off-line activities and games for all ages*. Canterbury, UK: Computer Science Unplugged.

Bers, M., Ponte, I., Juelich, K., Viera, A., & Schenker, J. (2002). Teachers as designers: Integrating robotics in early childhood education. In *Information technology in childhood education* (pp. 123–145). Morgantown, W. Va: AACE.

Bers, M. U. (2008). *Blocks to robots: Learning with technology in the early childhood classroom*. New York: Teachers College Press.

Bers, M. U. (2012). *Designing digital experiences for positive youth development: From playpen to playground*. Cary, NC: Oxford University Press.

Bers, M. U. (2018). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. New York: Routledge Press.

Bers, M. U., Seddighin, S., & Sullivan, A. (2013). Ready for robotics: Bringing together the T and E of STEM in early childhood teacher education. *Journal of Technology and Teacher Education, 21*(3), 355–377.

Bettinger, E., Ludvigsen, S., Rege, M., Solli, I. F., & Yeager, D. (2018). Increasing perseverance in math: Evidence from a field experiment in Norway. *Journal of Economic Behavior & Organization, 146*, 1–15.

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th workshop on primary and secondary computing education* (pp. 65–72).

Bravo Sánchez, F. Á., González Correal, A. M., & Guerrero, E. G. (2017). Interactive drama with robots for teaching non-technical subjects. *Journal of Human-Robot Interaction, 6*(2), 48–69.

Brosterman, N. (1997). *Inventing kindergarten*. New York: Henry N. Abrams.

Busteed, B., & Sorenson, S. (2015, August 20). *Many students lack access to computer science learning.* Retrieved from: www.gallup.com/education/243416/students-lack-access-computer-science-learning.aspx

Cejka, E., Rogers, C., & Portsmore, M. (2006). Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school. *International Journal of Engineering Education, 22*(4), 711.

Clements, D. H. (1999). The future of educational computing research: The case of computer programming. *Information Technology in Childhood Education Annual, 1*, 147–179.

Corbett, C., & Hill, C. (2015). *Solving the equation: The variables for women's success in engineering and computing*. Washington, DC: American Association of University Women.

Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014, November). Introducing teachers to computational thinking using unplugged storytelling. In *Proceedings of the 9th workshop in primary and secondary computing education* (pp. 89–92). Association for Computing Machinery (ACM).

Denworth, J. (2019). Debate arises over teaching "growth mindsets" to students. *Scientific American.* Retrieved from https://www.scientificamerican.com/article/debate-arises-overteaching-growth-mindsets-to-motivate-students

Dweck, C. S. (2002). The development of ability conceptions. In *Development of achievement motivation* (pp. 57–88). Cambridge, MA: Academic Press.

Dweck, C. S. (2008). *Mindset: The new psychology of success*. New York: Random House Digital Inc.

Elkin, M., Sullivan, A., & Bers, M. U. (2016). Programming with the KIBO robotics kit in preschool classrooms. *Computers in the Schools, 33*(3), 169–186. https://doi.org/10.1080/07380569.2016.1216251

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

Greiff, S., Wüstenberg, S., Csapó, B., Demetriou, A., Hautamäki, J., Graesser, A. C., et al. (2014). Domain-general problem solving skills and education in the 21st century. *Educational Research Review, 13*, 74–83.

Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2018). A scoping review of studies on computational thinking in K–12 mathematics classrooms. *Digital Experiences in Mathematics Education, 4*(1), 48–69.

Kazakoff, E., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal, 41*(4), 245–255. https://doi.org/10.1007/s10643-012-0554-5

Kim, J. O., & Kim, J. (2020). Development and application of art based STEAM education program using educational robot. In *Robotic systems: Concepts, methodologies, tools, and applications* (pp. 1675–1687). Hershey, PA: IGI Global.

Kuh, L. P. (Ed.). (2014). *Thinking critically about environments for young children: Bridging theory and practice*. New York: Teachers College Press.

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

Li, Y., & Bates, T. (2017). *Does growth mindset improve children's IQ, educational attainment or response to setbacks? Active-control interventions and data on children's own mindsets*. Edinburgh, UK: University of Edinburgh. https://osf.io/preprints/socarxiv/tsdwy/

Madill, H., Campbell, R. G., Cullen, D. M., Armour, M. A., Einsiedel, A. A., Ciccocioppo, A. L., et al. (2007). Developing career commitment in STEM-related fields: Myth versus reality. In R. J. Burke, M. C. Mattis, & E. Elgar (Eds.), *Women and minorities in science, technology, engineering and mathematics: Upping the numbers* (pp. 210–244). Northampton, MA: Edward Elgar Publishing.

Markert, L. R. (1996). Gender related to success in science and technology. *The Journal of Technology Studies, 22*(2), 21–29.

McClure, E. R., Guernsey, L., Clements, D. H., Bales, S. N., Nichols, J., Kendall-Taylor, N., et al. (2017). STEM starts early: Grounding science, technology, engineering, and math education in early childhood. In *Joan Ganz Cooney Center at Sesame Workshop*. New York: Joan Ganz Cooney Center at Sesame Workshop.

Messer, D., Thomas, L., Holliman, A., & Kucirkova, N. (2018). Evaluating the effectiveness of an educational programming intervention on children's mathematics skills, spatial awareness and working memory. *Education and Information Technologies, 23*(6), 2879–2888.

Metz, S. S. (2007). Attracting the engineering of 2020 today. In R. J. Burke, M. C. Mattis, & E. Elgar (Eds.), *Women and minorities in science, technology, engineering and mathematics: Upping the numbers* (pp. 184–209). Northampton, MA: Edward Elgar Publishing.

Moomaw, S., & Davis, J. A. (2010). STEM comes to preschool. *YC Young Children, 65*(5), 12.

National Science and Technology Council. (2018). *Charting a course for success: America's strategy for STEM education*. Washington, DC. Retrieved from: www.whitehouse.gov/wp-content/uploads/2018/12/STEM-Education-Strategic-Plan-2018.pdf

Nicholson, S. (1972). The theory of loose parts, an important principle for design methodology. *Studies in Design Education Craft & Technology, 4*(2), 5–14.

Perlman, R. (1976). *Using computer technology to provide a creative learning environment for preschool children*. Cambridge, MA: MIT, A.I. Laboratory.

Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., & Silverman, B. (1998, January). Digital manipulatives: New toys to think with. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 281–287). United States: ACM Press/Addison-Wesley Publishing Co.

Rodriguez, B., Rader, C., & Camp, T. (2016, July). Using student performance to assess CS unplugged activities in a classroom environment. In *proceedings of the 2016 ACM conference on innovation and technology in computer science education* (pp. 95–100). New York, NY, United States: Association for Computing Machinery

Scherer, R. (2016). Learning from the past–the need for empirical evidence on the transfer effects of computer programming skills. *Frontiers in Psychology, 7*, 1390.

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology, 111*(5), 764.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies, 22*(2), 469–495.

Shapiro, D. (2014, August). *Robot turtles: The board game for little programmers*. www.kickstarter.com/projects/danshapiro/robot-turtles-the-board-game-for-little-programmer/posts/597883

Smith, N., Allsop, Y., Caldwell, H., Hill, D., Dimitriadi, Y., & Csizmadia, A. P. (2015, November). Master teachers in computing: What have we achieved?. In *Proceedings of the Workshop in Primary and Secondary Computing Education* (pp. 21–24).

Steele, C. M. (1997). A threat in the air: How stereotypes shape intellectual identity and performance. *American Psychologist, 52*, 613–629.

Strawhacker, A. L., Lee, M. S. C., & Bers, M. U. (2017). Teaching tools, teachers' rules: Exploring the impact of teaching styles on young children's programming knowledge in ScratchJr. *International Journal of Technology and Design Education*. https://doi.org/10.1007/s10798-017-9400-9

Sullivan, A., & Bers, M. U. (2015). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*.

Sullivan, A., & Bers, M. U. (2017). Dancing robots: Integrating art, music, and robotics in Singapore's early childhood centers. *International Journal of Technology and Design Education.* Advance online publication. doi:https://doi.org/10.1007/s10798-017-9397-0

Sullivan, A., Bers, M. U., Mihm, C. (2017). Imagining, playing, & coding with KIBO: Using KIBO robotics to Foster computational thinking in young children. In *Proceedings of the international conference on computational thinking education.* Wanchai, Hong Kong.

Sullivan, A. A. (2019). *Breaking the STEM stereotype: Reaching girls in early childhood*. Lanham, MD: Rowman & Littlefield Publishers.

U.K. Department for Education. (September, 2013). *National curriculum in England: Computing programmes of study. Statutory guidance*. London: Crown copyright.

U.S. Department of Education. (2010). *Transforming American education: Learning powered by technology.* Washington, DC. Retrieved from: www.ed.gov/technology/netp-2010

Watson, E. (2020). STEM or STEAM?: The critical role of arts in technology education (and the critical role of art in technology). *Irish Journal of Academic Practice, 8*(1), 8.

Wyeth, P. (2008). How young children learn to program with sensor, action, and logic blocks. *The Journal of the Learning Sciences, 17*(4), 517–550.

Yakman, G. (2008, February). STEAM education: An overview of creating a model of integrative education. In *Pupils' attitudes towards technology (PATT-19) conference: Research on technology*. Salt Lake City, Utah: Innovation, Design & Engineering Teaching.

Yeager, D. S., Hanselman, P., Walton, G. M., Murray, J. S., Crosnoe, R., Muller, C., et al. (2019). A national experiment reveals where a growth mindset improves achievement. *Nature, 573*(7774), 364–369.

Zviel-Girshin, R., Luria, A., & Shaham, C. (2020). Robotics as a tool to enhance technological thinking in early childhood. *Journal of Science Education and Technology, 29*, 294–302.