Chapter 7
Computational Thinking and Young Children: Understanding the Potential of Tangible and Graphical Interfaces

Amanda Sullivan
Tufts University, USA

Marina Umaschi Bers
Tufts University, USA

ABSTRACT

Over the past few years, new approaches to introducing young children to computational thinking have grown in popularity. This chapter examines the role that user interfaces have on children’s mastery of computational thinking concepts, programming ability, and positive interpersonal behaviors. It presents two technologies designed specifically for young children: the KIBO robotics kit and the ScratchJr programming application, both of which focus on teaching young children introductory computational thinking skills in a cognitively and socio-emotionally developmentally appropriate way. The KIBO robotics kit engages children in learning programming by using tangible wooden blocks (no screens or keyboards required). ScratchJr also teaches foundational programming, but using a graphical language on a tablet device. This chapter presents examples of how each tool can be used in classroom settings and the potential benefits and drawbacks of each interface style. Suggestions for implementing each technology in a developmentally appropriate way are presented.

INTRODUCTION

Computational literacy has become “the new literacy” of concern to parents, educators, and caregivers of young children. As such, there has been a recent surge in the number of new tools and technologies being created for young users under the age of eight to explore computational thinking, coding, and engineering. This is evident based on the five-fold increase in ownership of tablet devices such as...
iPads, from 8% of all families in 2011 to 40% in 2013 (Common Sense Media, 2013) and the number is continually rising. In addition to screen-based or “graphical” technologies, new “tangible technologies” such as robotics kits, have also been growing in popularity with young children during the past few years. Tangible interfaces are of special interest in early childhood education as they resonate with traditional learning manipulates, such as Montessori blocks and Froebel’s specifically designed to learn about mathematical concepts (Bers 2008).

Unlike traditional graphical user interfaces (GUI), a tangible user interface (TUI) allows the user to input digital information by manipulating a physical object rather than using a screen, keyboard, or mouse (Strawhacker & Bers, 2015). Piagetian theory of the early 20th century provides a theoretical basis for promoting their use by proposing that interaction with the physical world is a primary learning avenue for young children (Piaget 1959). However, despite the new crop of tangible interfaces becoming available on the commercial market, more research is needed to understand their potential for promoting learning, and what children gain from tangible programming languages as compared to traditional graphical languages (Strawhacker & Bers, 2015). Teachers, parents, and researchers must carefully consider the benefits and drawbacks of these different interfaces. For example, while tangible interfaces may allow young children to develop fine motor skills through building and constructing, they may also be more expensive and harder for teachers to manage and store materials.

Research on developmental learning theory has shown that different tools and experiences may make different concepts easier for a learner to ingrain (Manches and Price 2011; Strawhacker & Bers, 2015). Using hands and objects instead of mental models may actually change the way a child remembers and retrieves the information taught (Strawhacker & Bers, 2015). The goal of this chapter is to introduce several examples of graphical and tangible interfaces for teaching computational thinking and coding to young children in order to expose the reader to the range of tools available to young children. We will provide in-depth descriptions of KIBO (a tangible robotics kit for young children ages 5-7) and ScratchJr (a graphical programming language for young children ages 5-7). We use KIBO and ScratchJr to highlight the critical differences between graphical and tangible interfaces for young children and the benefits and drawbacks of each. Finally, practical considerations for educators, parents, and researchers to consider when choosing tools are presented.

**Literature Review**

**Computational Thinking in Early Childhood**

The term “computational thinking” has been defined in many ways and encompasses a broad range of analytic and problem-solving skills, dispositions, habits, and approaches used in computer science (Barr & Stephenson, 2011; International Society for Technology Education and The Computer Science Teachers Association, 2011; Lee et al., 2011). Computational thinking can be thought of as solving problems algorithmically and developing a sense of technological fluency (Bers, 2017; Bers, 2010; Papert, 1980) Children as young as four years old can learn foundational computational thinking concepts (Bers, 2017; Bers, 2008) and this kind of learning can support their literacy, mathematical, and socio-emotional development (Kazakoff & Bers, 2012; Kazakoff, Sullivan, & Bers, 2013).

While computational thinking is rooted in computer science, many have argued that it is a universally applicable attitude and skillset that fundamental for everyone to master, just like reading, writing,