Cognitive Guidelines for the Design and Evaluation of Early Mathematics Software: The Example of *MathemAntics*

Herbert P. Ginsburg, Azadeh Jamalian, and Samantha Creighan

Introduction

This chapter shows how cognitive psychology can inform the design and evaluation of software for early mathematics education, and how the resulting software can provide new approaches to evaluation of learning and to basic cognitive research.

The need for improved early mathematics education is abundantly clear. Too many children (at least in the U.S.) perform poorly in mathematics from the earliest days of school (Mullis et al. 1997). The problem is especially acute in the case of underprivileged, low-income children, who start behind (Starkey and Klein 2008) and fall further behind as they grow older (Cunha et al. 2006; Duncan et al. 2007).

Although there are many contributors to low academic achievement, ranging from poverty to teacher pay, the primary factor is *not* young children's inability to learn mathematics. A very large body of research shows that children naturally develop a surprisingly proficient and complex "everyday mathematics" (Sarama and Clements 2009b), which provides a useful foundation on which mathematics education can build (Baroody 2004). The research also shows that quality early mathematics education can have long term positive impacts on achievement and can provide substantial benefits for those who need the most help, namely underprivileged, low-income children (Cross et al. 2009).

We propose that the affordances of computer technology, although hardly a panacea, offer the possibility of transformative improvements in early mathematics education. It is possible and desirable, we argue, to design software to help children learn mathematics, to help teachers teach it, and to eliminate the need for textbooks as we know them.

Our optimism (some concerns will follow) is fueled by two developments. One is that education authorities now accept that computer technology has an important place in education from elementary school through postgraduate studies. The

H.P. Ginsburg (✉) · A. Jamalian · S. Creighan
Teachers College, Columbia University, New York, NY, USA
e-mail: hpg4@columbia.edu

L.D. English, J.T. Mulligan (eds.), *Reconceptualizing Early Mathematics Learning*, Advances in Mathematics Education, DOI 10.1007/978-94-007-6440-8_6,

second, and more important, is that touch screen devices, notably the iPad, have become ubiquitous in a short period of time. Software developers have been releasing very large numbers of mathematics apps for young children from age 3 upwards. In July 2012 there were more than 20,000 education apps designed for the iPad, and more than 1.5 million iPads in U.S. schools (Brian 2012). It is not uncommon to see very young children using touch screen devices with reasonable proficiency. Even toddlers navigate the screen, access their favorite app icons, and play with the software for long periods of time.

Although only in their infancy (like some of their users), the new touch screen devices seem ideal for the population on which this paper focuses: young children from roughly age three to six whose reading ability is limited or non-existent and who may be unable to move a mouse with facility but can nevertheless touch and manipulate attractive virtual objects on a brilliantly vivid screen. Touch screen devices that can talk to young children (when they tap on a word or numeral or perform certain actions) and that allow them to touch and manipulate virtual objects can set the stage for—but definitely do not guarantee—dramatic advances in the quality of software.

Yet there is reason for concern. The availability of large numbers of apps does not in itself solve the educational problem. Individual apps by the hundreds or thousands seem to emerge with no evident plan, rhyme or reason. Some of the most popular apps have a limited focus. For example, Space *Math* (Mclean 2012) drills students on number facts but is little more than an efficient worksheet that does not promote conceptual knowledge. Although drill can be useful, there seem to be few examples of mathematics software for young children that promote conceptual understanding and non-trivial problem solving.

Further, although rigorous evaluations are rare, our informal observations suggest that the quality of mathematics apps and other software is generally not impressive. Many designers and publishers claim that their software is of high quality because it is "research-based." Yet these assertions need to be taken with a very large grain of salt. Educational researchers have argued that research is seldom used in meaningful and effective ways in software development (Sarama and Clements 2002). There are exceptions, an example of which is *The Number Race*, designed for 7- to 9-year-olds with mathematical difficulties, which focuses on improving number sense through number comparison tasks (Wilson et al. 2006a, 2006b). Another exception is *Dots2Track* (Butterworth and Laurillard 2010), which also targets children with mathematics learning difficulties. Both of these software programs derive from a research-based analysis of children's learning difficulties, and show promise of correcting cognitive deficiencies and promoting mastery of early mathematics.

We propose that cognitive science (which for purposes of this paper we define as including contributions from the overlapping and sometimes vaguely defined disciplines of cognitive developmental psychology, educational psychology, learning science, cognitive psychology, and mathematics education) can and should play an essential role in exploiting the powerful affordances of computers in the service of education. First, cognitive principles can provide a framework for the design of educationally rich software, ensuring that its pedagogy effectively promotes mathematical proficiency in the broad sense, including both the motivated learning of rich content and the development of genuine mathematical thinking. Such software can help teachers to change their roles and improve their teaching, and eventually will make traditional textbooks obsolete. Second, the cognitive principles can contribute to the usability testing and meaningful evaluation of software designs. The cognitive principles can also guide the formative and summative assessment of children's learning, as well as the evaluation of achievement and the effectiveness of software. And finally software based on cognitive principles can return the favor by setting the stage for new kinds of basic research into children's mathematics learning in rich environments.

Cognitive Principles for the Design of Software

We begin with what the computer can do. We know that software affordances can support learning (Sawyer 2006) in several ways. Computers can represent abstract knowledge in interactive visual models; touch screens allow children to manipulate virtual objects; computer tools provide opportunities for learners to explore and develop solutions to interesting problems; and computers allow building a community of learners and collaborative learning. Software can also engage children in "microworlds"—artificially designed "mathemagenic" environments that entail and stimulate the exploration of mathematical ideas, foster the development of thought and skill, and offer children powerful tools to do significant mathematics (Hoyles and Noss 2009; Hoyles et al. 2002; Papert 1980). Microworlds can include goal-driven activities, virtual objects, tools, representations, scaffolds, feedback, pedagogical agents, interaction, fantasy, challenges, communication and collaboration, record keeping and reporting.

We propose 6 cognitive design principles that can exploit these affordances.

- Engage children in cognitively and mathematically appropriate activities.
- Develop effective models for representing abstract ideas.
- Encourage accurate and efficient strategies.
- Identify and eliminate bugs and other misconceptions.
- Design appropriate physical interactions.
- Integrate narratives and stories with mathematical concepts.

For each design principle, we outline important issues that software developers need to consider, and then we discuss examples of how cognitive principles can inform design.

Engage Children in Cognitively and Mathematically Appropriate Activities

The first step in designing learning activities is to clearly define the content to be learned and taught. The new Common Core State Standards Initiative (2010) provides broad guidance, based extensively on cognitive developmental research. For example, the Kindergarten Standards propose that children should learn the number names and count sequence to 100; count to tell the number of objects; and compare numbers.

To those unfamiliar with cognitive psychology, teaching this content would appear to be easy. After all, what can young children learn beyond a little counting, names of shapes, and memorized facts? But neither the mathematical content nor its learning and teaching are simple. Software developers have produced many apps and programs designed to help children learn these important topics. Unfortunately too many apps focus on the rote teaching of number words and simple enumeration without regard to its meaning (for example, *Toddler Counting*, iTot Apps, LLC 2011). Although knowing the number words and being able to recite the list in the correct order is important, children need to learn a great deal more about number than current software apps teach. Rich content should include thinking as well as facts and procedures.

To develop effective software that can help implement the Core Standards, designers need to understand the specific cognitive processes and obstacles involved in young children's learning the counting words, enumeration (determining the number of a set of objects), and number comparisons of various types. And it goes without saying that designers need to have a deep knowledge of the mathematics itself, which in the case of young children is far from trivial: it deals with fundamental concepts of number theory.

Fortunately, a substantial body of research can provide guidance for software development in the area of early number. Gelman and colleagues (Cordes and Gelman 2005; Gelman and Gallistel 1978; Gelman 1993; Gelman and Gallistel 1986) show that learning to enumerate involves several components. The child must acquire three "how-to-count" principles: (1) stable order principle, (2) the one-to-one correspondence principle, and (3) the cardinal principle. The Stable Order Principle requires that symbols have a consistent order across counting occurrences. The One-to-one Correspondence Principle explains that for every object in the counting set only one and one counting symbol is applied. The Cardinal Principle refers to the fact that the last symbol of a count represents number of objects in the set that has been counted. A child who counts 6 candies as "1, 2, 3, 4, 5, 6" and says, "I have 6 candies," knows the Cardinal Principle, at least for the number 6. Gelman and Gallistel (1986) also define two "what-to-count" principles: (1) the Abstraction Principle states that any combination of discrete objects could be counted (e.g. a set of heterogeneous farm animals or abstract entities such as months in a year), (2) the Order Irrelevance Principle states that a set can be counted in any order (left to right, right to left, or any other order) and yet the cardinality of the set does not change.

Children also need to understand the meaning underlying the distinctive symbolic language of mathematics and more generally to appreciate that mathematics is meaningful. Vygotsky (1978) proposed that early mathematics education should help children learn to synthesize their "spontaneous concepts" (everyday mathematics) with "scientific concepts" (the organized, formal mathematics that constitutes the accumulated cultural wisdom and that schools try to teach). "The strength of scientific concepts lies in their conscious and deliberate character. Spontaneous concepts, on the contrary, are strong in what concerns the situational, empirical, and practical" (Vygotsky 1986, p. 194). This approach, similar to Dewey's (1976), aims to integrate the best of the vital and spontaneous with the best of the rigorous and scientific so as to produce meaningful knowledge. Another way of saying this is we should help children to mathematize their meaningful everyday mathematics.

To create useful software, designers need to draw upon cognitive research to develop clear answers to the following three questions relating to content:

1. What should be the specific mathematical and cognitive content of the activity?
2. What is the developmental trajectory for mastering the concepts and skills in question?
3. What prior knowledge is required to master understanding of the new concept?
4. How does the software promote understanding of symbols?

To illustrate how this can be done, we consider one set of *MathemAntics* activities designed to help pre-school children learn about numerical relations and comparison of set sizes. To design the content for the activities, we first analyzed the developmental trajectory for mastering the number relation concepts.

2- and 3-year-old children begin to learn relations such as *more* and *same* (Clements and Sarama 2007, 2008; Fuson 1992a, 1992b; Ginsburg 1989). Children can compare two sets using subitizing (*seeing* the number immediately without counting), length, or density strategies (Mix 1999). We based the content of the simplest *Equivalence* activities in *MathemAntics* on what we know about children's knowledge, and synthesized activity content with their mathematical understanding. The simplest activity encourages the child to visually inspect two sets of 3 or fewer objects arranged in a row or in a column, with the length of the rows congruent with the cardinal values. In higher levels, larger sets with numbers of objects beyond the subitizing range are presented to the child for comparison, and the child may still use length or density strategies to compare these sets. In the *Hens Laying Eggs* activity, for example, children are asked whether Fluffy and Fancy Pants have the same number of eggs (see Fig. 1). The eggs for each hen are arranged in a row, and the length of each row is congruent with the cardinality of that row (i.e. the set with fewer eggs is arranged in a shorter row).

Another simple activity called *Bedtime* (see Fig. 2) encourages children to compare two sets by matching objects from one set to the other set. Children put each animal in a bed to help it get ready to sleep, and ring a bell when they are done. The one-to-one activity not only emphasizes the matching strategy but also highlights the one-to-one counting principle.

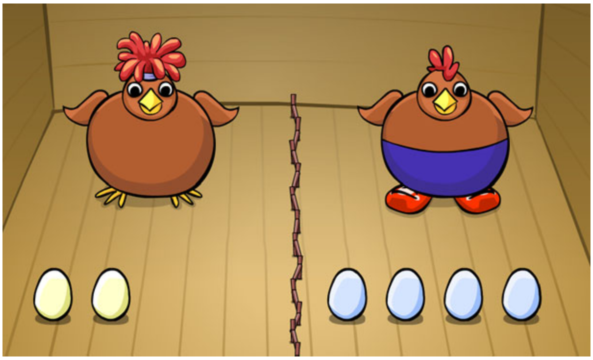


Fig. 1 Level 1 of Hens Laying Eggs activity

Subitizing, length, or density strategies are not efficient for comparing two sets that are larger than 3, have close to a 1:1 ratio, or have incongruent physical and numerical attributes (e.g., two equal sets with different densities). By the age of 4, children should be able to use counting and matching strategies to compare the size of sets with 5 or fewer objects (Cross et al. 2009). To use counting successfully, children need to be able to count each set accurately, remember the cardinality of the first set while counting the second set, and then compare the cardinalities of the two sets. Fluency in counting is important. Without it children may forget their first count result by the time they have counted the second set. Children also need to know order relations of cardinal numbers—the further numbers are along in their counting list, the larger quantities they represent (e.g., Fuson et al. 1982). Therefore, fluency in counting (i.e. mastering all 5 counting principles) is the required prior knowledge for using counting to compare two sets. For example, to compare the two schools of fish in the *Pop the Bubble* activity, children need to count each set separately and pop the bubble that has more fish in it (see Fig. 3).

In addition to counting fluency and knowledge of order relations of cardinal numbers, executive function skills are critical in the successful use of counting strategies. Inhibitory skills, working memory, and cognitive flexibility (Diamond 2008) are of particular importance. To compare two sets, children need to be able to count one set, stop, and start from number 1 to count the other set. In fact, in our own usability research we observed that some 4 year olds fail to stop after counting the first set and continue counting the second set together with the first set. Children who are able to count two sets separately may also fail to compare the two sets based on their counting results due to working memory limitation: by the time they count the second set they may forget the counting result for the first set. To assist their memory and to teach them the meaning of the symbol, we provide the option to have the numeral for each set written at the top of the bubble. So, when the child indicates how many fish are in a bubble, the numeral will appear on top of the bubble and stay on the screen (see Fig. 4). This helps the child to learn to read the written numerals and understand what they mean.

By the age of 5, children should be able to use counting strategy to find out which is more and which is less for two numbers ≤ 10 (Cross et al. 2009). In the Pop the Bubble activity for example, we ask children to compare two sets of fish and pop

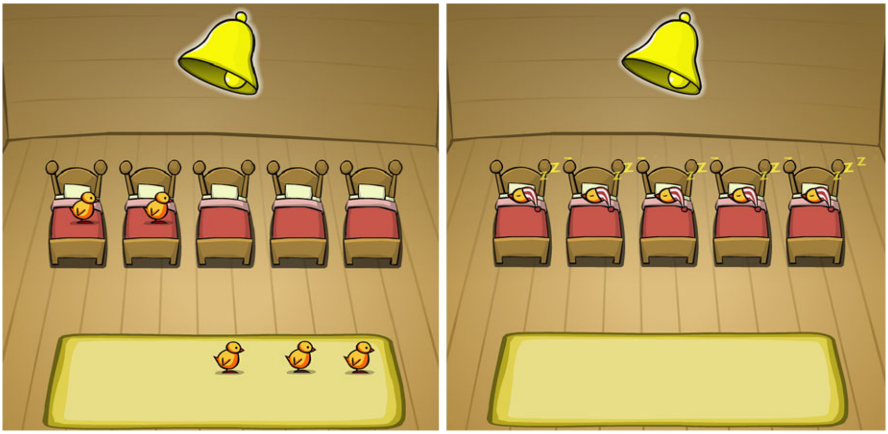


Fig. 2 Bedtime activity

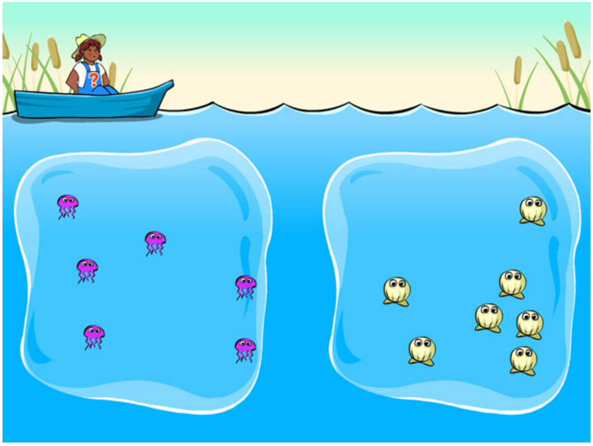
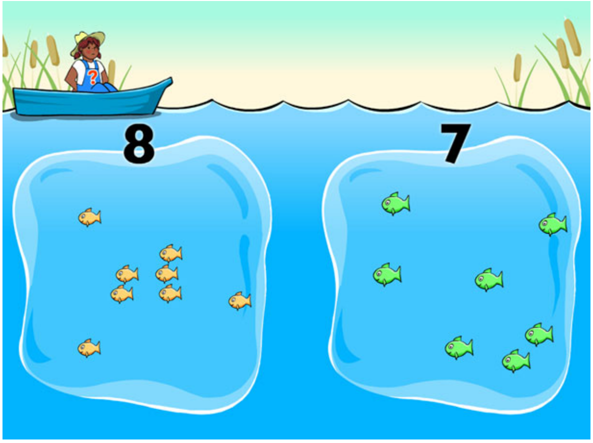


Fig. 3 Pop the Bubble activity

the bubble with fewer or more fish in it. At this age, children also need to learn the meaning of the equal and non-equal symbols, and we introduce these symbols as part of the activities so that children will be able to achieve a synthesis between their everyday knowledge (this group has more than that) and the appropriate symbol. For example, in the Quick Compare activity, we ask the child to compare number of animals in two barns by clicking on the equal or non-equal signs (see Fig. 5).

By first grade, not only do children need to know which set has more, but they should know "how many more", or "fewer" objects one set has compared to another (Cross et al. 2009). For example, in more difficult levels of the Hens Laying Eggs activity we ask children to make Fluffy have 2 more eggs than Fancy Pants. The child may remove eggs by clicking on them, or touch a chicken's belly to have her lay an egg. At this age, children are again introduced to "more than" or "fewer than" symbols and number sentences in the context of their own everyday mathematics.

Fig. 4 Pop the Bubble activity with number symbols



Conclusion We have seen that cognitive psychology provides detailed and specific information concerning the cognitive content of the mathematics to be learned—cardinality, number comparison, and the rest—and the struggles children have in learning it. Key learning and teaching principles should inform the design of software. For example, to produce a meaningful synthesis between informal knowledge and the formal symbolism of mathematics, we linked the eggs in the Hens Laying Eggs activity with the number sentence (8 > 6 in Fig. 6).

Fig. 5 Quick Compare activity

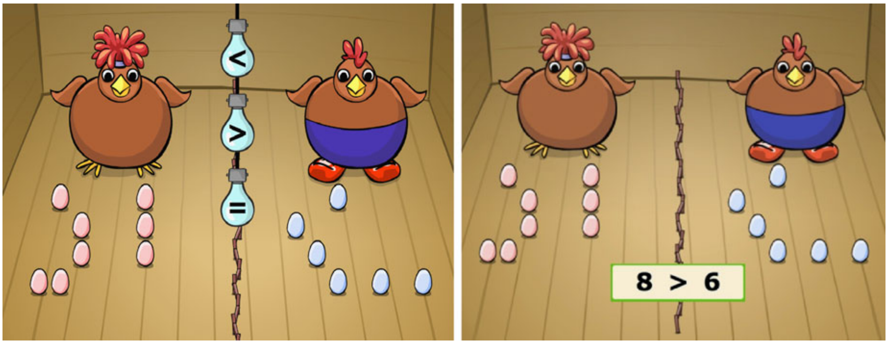


Fig. 6 Hens Laying Eggs activity with number sentence

Develop Models Representing Abstract Ideas

Perhaps the most important educational goal is to promote deep understanding of mathematical principles. Mathematics educators use a variety of instructional aids to help children learn abstract concepts. These aids can take the form of visual representations, models, tools, or manipulatives. Whatever the name, the goal is the same—to represent a mathematical concept or relationship in a meaningful way. Many mathematical ideas can be modeled through such representations, such as representing the number five as five objects, a block with five discrete units, or a continuous line shading five units on a number line. The way in which a concept is modeled can directly influence the way the child conceptualizes and understands the mathematics. For example, modeling the multiplication problem 3×4 as three equal groups of four is different than representing the problem as the area of a rectangle that is 3 units long and 4 units wide. Though an adult understands that both of these representations will yield the same total, 12, a child new to the concept may struggle to synthesize multiple representations. And indeed the two representations are in fact different. The rectangle representation can be used to deal with continuous area while the group of objects representation must involve discrete quantities.

A powerful way to teach children about abstract mathematical ideas is through manipulatives embodying various kinds of representations (Mix 2010). The basic idea is that manipulatives—things that hands can touch, feel and move—can help students to create mental representations of mathematical ideas and procedures. Manipulatives are not just things to play with, but artifacts designed to help the learner construct sound ideas. Manipulatives are successful when they can be abandoned because they are no longer needed to support the ideas.

Used properly, manipulatives can advance our general educational principles. They are responsive to the child's current cognitive state in that they may involve, for example, manipulating sets of objects instead of symbolic statements. Manipulatives elaborate on everyday knowledge by helping the child to perfect judgments of numerical magnitude (for example, by providing practice in comparing the numbers of sets of objects). Activities involving manipulatives can help the child mathematize by showing the numerals corresponding to various numbers of objects (for example, the numeral 10 next to a collection of 10 blocks). Manipulatives can also promote a synthesis of everyday mathematics (this is a much taller tower of blocks than that tower) with the symbolic (this tower has 13 blocks but that tower has only 5).

Concrete manipulatives like Cuisenaire Rods or Unifix Cubes have been used for a long time to teach fundamental concepts of number. Virtual manipulatives (as defined by Moyer et al. 2002) and models can also help visually represent mathematical ideas and relationships (Mix 2010). Indeed computer technology can be used to create virtual manipulatives that in some ways may be more powerful than their concrete counterparts. Although it is virtually impossible to have the child work with 5,000 blocks, doing so can be child's play on the computer screen. The representation can be visual or pictorial (as when a child sees a randomly arrayed collection of animals on the screen), which can be manipulated by a mouse or by the fingers on a touch screen (for example, into groups of 3 animals in a line), and can be connected in real time to symbolic representations (for example, if the line is 3 long the numeral is 3, but if an object is added to the line the numeral changes to 4) (Moyer-Packenham et al. 2008). To fully understand and embody the mathematics, a child must be able to flexibly connect and utilize multiple representations, understanding that a virtual group of five chicks can be represented by a five block (comprised of 5 connected unit blocks) and by the numeral 5. If the child sees the numeral 5, she should be able to represent it with the 5 block or the 5 virtual chicks. This is the kind of flexible synthesis entailed in deep understanding.

Although work with manipulatives and models is widespread and often acclaimed as beneficial, research regarding their effectiveness shows that their use does not necessarily ensure deep understanding (Moyer-Packenham et al. 2008; Mix 2010). The issue is not simply whether or not manipulatives are used. It should come as no surprise that the fundamental question seems to lie in the circumstances in which particular models and manipulatives can be used most effectively for particular purposes. Moreover, children need enough time to interact with a specific representation to fully understand the underlying mathematical content, must be carefully introduced to new representations, and must be provided with scaffolds to connect multiple representations in meaningful ways (Sarama and Clements 2009a; Mix 2010).

Unfortunately, current software seldom uses virtual representations in meaningful ways. Some utilize multiple representations for the same concept but lack the scaffolding to promote synthesis. Others provide representations that are conceptually confusing (for example, representing one as eight fish forming the numeral one!) or are inappropriate for the intended age (having instructions appear as written text only for preschoolers!). Other mathematics software programs stress rote learning of number facts and make almost no use of virtual manipulatives that offer useful representations.

Given the potential importance of virtual manipulatives, designers need answers to the following three questions:

1. What models can be used to represent a concept?

2. What are their possible benefits?
3. How can they be connected with other models?

What Models Can be Used to Represent a Concept? When designing educational software, designers must first consider the different ways in which the concept could be modeled. Inspiration may be drawn from existing school curricula or from the cognitive psychology literature. The most effective models clearly and appropriately represent and help the child form a useful mental representation of the abstract mathematical concept. For example, the multiplication problem 3×4 can be represented as an array of 3 rows of 4 dots each; a rectangular array of 3 by 4 squares; 3 jumps of length 4 on a number line; and 3 plants with 4 flowers on each. These are all different ways of thinking about 3 times 4, and each has its specific strengths and weaknesses. As mentioned earlier, a discrete array cannot represent area, and neither can jumps on a number line, which in turn cannot represent the idea of "number of elements per unit" as well as the flowers can. Given these possible representations that are useful for different purposes, designers may be able to transform the appropriate model into a powerful virtual manipulative.

Benefits Although we have highlighted the importance of using multiple models, we will provide an in-depth look at one, the number line. The number line is versatile in that it can be thought of as a visual representation of number, but it can also model mathematical relationships (such as modeling how 80 is larger than 30), and serve as a mathematical tool to solve problems (such as solving $3 + 5$ by making jumps on a number line). Psychological research has shown that the number line is very powerful in representing mathematical concepts such as numeral identification, counting, and most importantly numerical magnitudes. Working with the number line can build number sense and contribute to the development of a mental number line (Jordan et al. 2006).

As children begin to study the symbolic number system, they may learn to map magnitudes onto a mental number line. They may know that 3, 4, and 5 are pretty close to one another, but that 100 is far away. However, their mental number line is imperfect. It follows a logarithmic representation for larger and unfamiliar quantities. In other words, children have a pretty good idea of where the smaller numbers fall on the line, but after a certain limit, the larger numbers tend to be jumbled together as very big. To examine the mental number line, researchers have used a task that involves giving a child to place various target numbers on a blank number line with only the endpoints marked (Booth and Siegler 2006; Laski and Siegler 2007; Moeller et al. 2008). Instruction and exposure to larger numbers, as well as feedback on performance on the number line estimation task, help children to shift from a logarithmic representation to a more accurate linear mental number line. Several interventions that focus on the number line—for example, Early Learning in Mathematics (Chard et al. 2008), The Number Race (Wilson et al. 2006a), and The Great Race (Ramani and Siegler 2008)—have reported successful outcomes. Siegler and Booth (2004) also found that instruction of this type may have long-term benefits in other areas of mathematics: the likelihood of using a linear representation of number is highly predictive of later mathematical achievement.

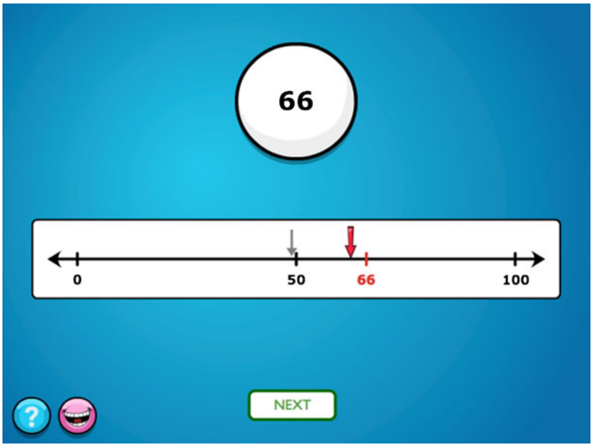


Fig. 7 Basic Number Line activity

When designing *MathemAntics* we aimed to use the number line as a powerful virtual manipulative. For example, in *MathemAntics*, as in the traditional number line estimation task, the child places various target numbers on a number line. But then the software gives the child multiple attempts and specific verbal and visual feedback, which has been shown to assist children in forming an accurate linear mental representation of number (Laski and Siegler 2007). And of course, the computer allows the educator to set various parameters such as the length of the line, the presence or absence of hash marks in any places.

Figure 7 shows an example of the basic number line activity in *MathemAntics*. The gray arrow denotes the child's incorrect first attempt; the red arrow denotes the child's current attempt; and the correct position of the target, 66, is shown in red as visual feedback.

We have also designed an activity that allows children to build their own number line by positioning numerals on the line. This activity scaffolds children to pay careful attention to the order and spacing of the numbers. When finished, the child can compare his number line to a correct number line above, and fix the former accordingly. This activity is intended to encourage children to explore and learn number relations.

Adjusting the parameters of the number line allows modeling of other important mathematical concepts such as number operations (modeling addition as jumps on a number line), place value (labeling the decades only on a 0–100 number line) (Moeller et al. 2008), measurement (Petitto 1990), and rational number (Schneider and Siegler 2010). The number line is a highly versatile model capable of conveying many mathematical concepts. However, designers should be aware that simply including a number line does not guarantee the child will master all of these concepts. This reiterates the importance of the first cognitive design principle, knowing the specific developmentally appropriate conceptual content. We will revisit this issue in the discussion of later principles and provide examples of how designers can use the number line in useful ways to encourage deep understanding.

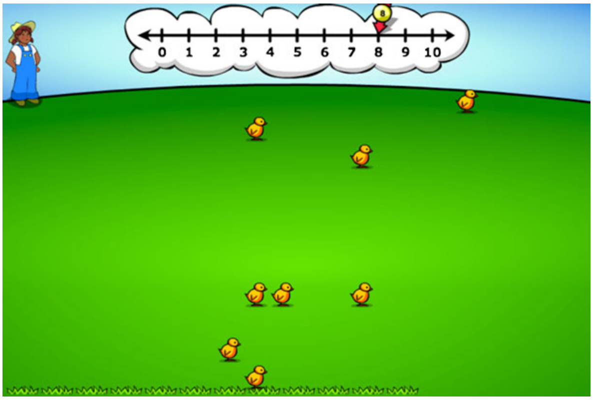


Fig. 8 Enumeration activity using the number line as response mechanism

Ways to Connect Multiple Representations As we have seen, the number line can be a useful tool for teaching many concepts, from order to rational number. It can also be used to integrate multiple representations, thus promoting the synthesis of different areas of mathematical knowledge.

To achieve this goal, the number line appears in other *MathemAntics* activities as a response mechanism or tool. In early enumeration activities, a child may explore the learning environment by adding or taking away chicks in a field. The total number of chicks is simultaneously represented with a yellow magnitude bar on a number line that changes in real time based on the child's actions. In a later enumeration activity, the child indicates his answer by clicking the corresponding numeral on the number line (see Fig. 8).

Feedback or scaffolding can highlight and count the animals out loud while a yellow magnitude bar represents the quantity on the number line, thus aiming to carefully connect the representations for the child. Once the child masters enumeration, the concrete representation of chicks in a field may be taken away leaving only the number line. In later activities, the child may choose to use various tools, like the number line, to solve problems involving mathematical operations. In this advanced case, the child has already been introduced to multiple representations, provided with appropriate scaffolds to connect them, and may now choose how to represent the problem.

Conclusion Many kinds of manipulatives can be used to represent mathematical ideas. As a virtual manipulative, the number line can be used to promote number sense and a more accurate mental number line; to teach ideas of order and rational number (as well as any of the operations on the whole numbers); and to promote a synthesis between the written numbers and important mathematical ideas.

Encourage Accurate and Efficient Strategies

Learning mathematics is not only about providing correct answers to problems, but also using efficient strategies to solve those problems. Strategies may be efficient or inefficient, accurate or inaccurate. One major goal of mathematics education is to encourage children to use increasingly effective strategies for solving problems.

When designing learning activities to teach new strategies, two points are important to consider. First, adoption of new strategies is often quite slow: children may continue using prior strategies even if newly adopted ones have clear advantages that the children themselves can explain (Siegler and Jenkins 1989; Alibali 1999; Chen and Klahr 1999; Goldin-Meadow and Alibali 2002; Granott 2002; Kuhn et al. 1995). Second, rate of adoption of new strategies varies in relation to two factors: accuracy and efficiency (Siegler and Svetina 2006). If the newly adopted strategy significantly improves accuracy relative to a prior strategy (generally from consistently incorrect to consistently correct), it will quickly replace the old one. If both the new and old strategies result in correct performance, children adopt new strategies that boost speed or dramatically reduce processing more quickly than do strategies with less substantial advantages. Thus, differences in both accuracy and efficiency of the new and old strategies contribute to the rate of adaptation, with accuracy being the more powerful factor.

Very few mathematics apps for children consider strategy use. Many apps employ a multiple-choice format and simply ask children to provide an answer. Although these apps may help children to practice and improve their mastery of number facts and calculation skills, effective mathematics education requires more apps that promote the adoption of effective strategies for problem solving.

Prior to designing an app to promote strategy learning, three questions should be answered:

1. What are the possible strategies to solve the problem?
2. Which strategies lead to more accurate answers and under what conditions?
3. What strategies are more efficient than the others for a specific problem?

To answer these questions, we draw upon findings of cognitive psychology, as well as our clinical interviews and observations during usability and play testing. Usability testing involves determining whether the user can do whatever he/she is supposed to do. For example, can a 3 year old drag objects on a touch screen? Play testing involves determining whether the child uses software as it was designed to be used. For example, if the software offers a potentially useful tool, does the child use it for the intended purpose? We then illustrate answers to these questions by considering the design of *MathemAntics* activities.

Different strategies may be employed to compare magnitudes of two sets depending on the features of the problem. Perceptual subitizing, length, or density strategies are the most basic. If the set sizes are less than 3 or 4, the child may simply subitize—quickly "see"—the number of objects in each set and evaluate if they are the same or one is more. For young children, fast subitizing is only efficient and accurate if the number of objects in each set is less than 4 (Sarama and Clements

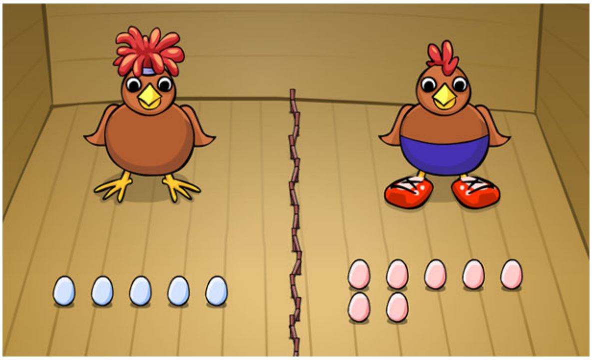


Fig. 9 Encouraging visual comparison in Hens Laying Eggs activity

2009b). When comparing large sets of objects (more than 4), children may employ visual comparison strategies employing length or density. Children judge a group of objects to be greater in number when it is either longer or denser than another group. Visual comparison strategies are efficient and relatively accurate if the sizes of objects in the two sets are similar, and the area occupied by each set is congruent to its numerical value (Mix 1999). Further, the visual comparison strategies are accurate within a ratio limit (Feigenson et al. 2004). Thus, it is easier to see the difference between 4 and 5 than between 94 and 95. However, comparing cardinality of two sets based on visual attributes such as length and density is not accurate if the two sets have a close to 1:1 ratio, or if the visual attributes are incongruent with the numerical property of the sets. Conceivably counting was developed to deal with relatively difficult problems like these.

Children may also use a matching strategy when comparing two relatively small sets. For example, a child may examine whether there are the same number of beds as there are animals by corresponding each animal with one bed. However, matching is not efficient when comparing 20 beds and 20 animals or when sets are not in close proximity. Counting is a more efficient and sophisticated strategy for comparing the quantities in situations where visual comparison strategies are not efficient.

Software has unique affordances to encourage the use of strategies by highlighting the advantages of new strategies or limiting the resources needed to use another strategy. Following are five design scenarios.

Visual Representations The visual properties of a problem can be set to encourage certain strategies over others. For example, to encourage visual comparison strategies to compare the quantity of two sets, objects in each set could be positioned in rows and columns inviting the child to compare the length of the rows: see Fig. 9.

As another example, to encourage use of the "count on" strategy to solve an addition problem, the bigger addend could be represented with a number, and the smaller addend with discrete objects, as in this hypothetical example (see Fig. 10).

Tools Tools can encourage the use of specific strategies to solve problems. For example, to encourage comparing quantity by matching, in *MathemAntics* children

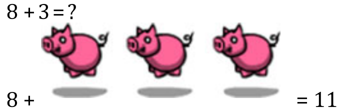


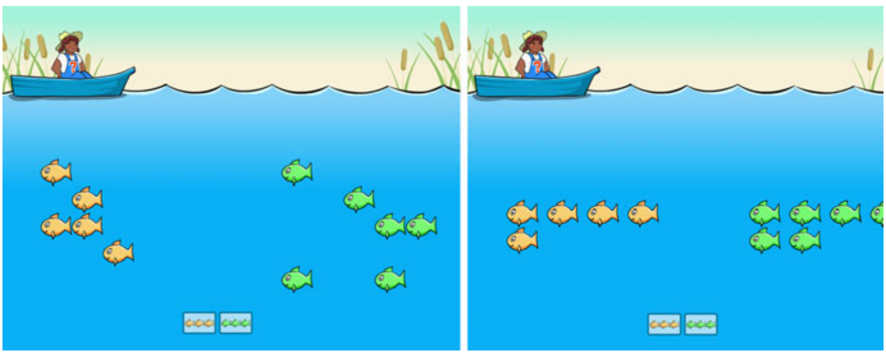
Fig. 10 Encouraging "Count On" strategy for addition

may have access to a "line-up" tool that arranges objects in a row and thus provides an organized visual comparison (see Fig. 11).

Note that the same tool that is designed to highlight a strategy may hinder appropriate use of another strategy. For example, if the activity aims to encourage comparing quantity by counting, the line-up tool should not be accessible. In higher levels of the activity, the "line-up" tool is no longer available, but the child may drag objects and arrange them in a certain way. To encourage counting, the dragging option could be made inaccessible. Thus, tools and virtual manipulatives can focus student attention on specific mathematical relations and processes in order to encourage one strategy over another, a method called "focused constraint" (Moyer-Packenham and Westenskow 2013).

Setting Parameters of the Problem Parameters—special features of a problem, like the number of objects or their spatial arrangement—can be adjusted to encourage certain strategies or discourage others. Petitto (1990) used number line estimation and specially designed rulers to explore the development of numerical proportion abilities and understanding of units of measurement. She found that in the number line estimation task children use many different (often inefficient) strategies that suggest limited understanding of proportional reasoning. For example, younger children often utilize counting strategies (either counting up from left to right or counting down from right to left), oftentimes with inaccurate units. By contrast, older children begin to correctly utilize the midpoint strategy when appropriate (e.g. counting on from 50 to get to 53 on a 0–100 line).

The specific parameters that can be manipulated in the MathemAntics version of the number line estimation task, such as providing tick marks of varying inter-



Before using the line-up tool After using the line-up tool

Fig. 11 Using the Line-Up tool to encourage visual comparison strategy

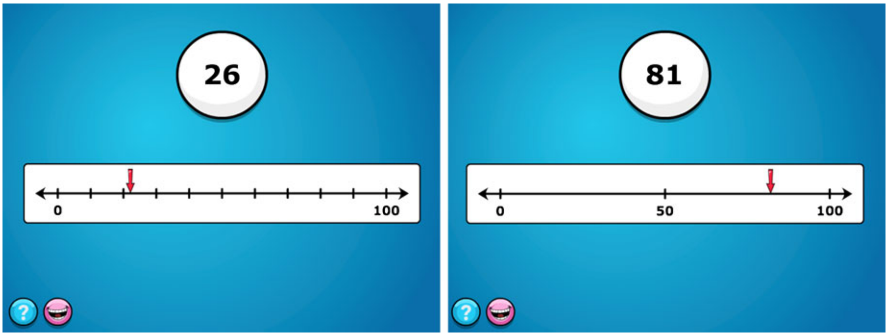


Fig. 12 Adjusting parameters of the number line to encourage different strategies

vals that can be labeled or not, may encourage the use of different effective strategies. However, the presence of tick marks may also encourage precise but tedious counting rather than estimation based on number sense. Further, showing the tick marks may prevent a child from shifting to using a more advanced midpoint strategy. Hence, the parameters of the number line may also be set so that the midpoint (rather than the tick marks) is always visible. In the left image of Fig. 12, we see a number line set to display tick marks encouraging precision and counting-based strategies whereas in the right image we see a number line with only the midpoint visible, encouraging the midpoint and more approximation-based strategies.

Feedback Simply indicating correctness or incorrectness of an answer may not be the best way to respond to a child's efforts. Instead, feedback may be designed to encourage a certain strategy. For example, if the goal of an activity is to urge children to compare sets by counting, the software feedback to an incorrect response could be "Oops, you were wrong this time. Let me show you how I know that. Fluffy has 1, 2, 3, 4—4 eggs altogether, but Fancy pants has 1, 2, 3, 4, 5—5 eggs altogether. They don't have the same number of eggs." On the other hand, if the activity aims to encourage comparing quantity by matching, the feedback could be a simple animation, matching eggs of the two hens and highlighting the leftovers.

Scaffolding Carefully designed scaffolds could also help a child in adopting a new strategy. Scaffolding may highlight the features of the problem that are relevant to a strategy, ease the difficulties involved in adaptation of a new strategy, and model the strategy for the child. These scaffolds may be implemented in the software as feedback to the child's wrong answer. The *MathemAntics* version of the number line activity provides scaffolds designed to help children to utilize certain strategies. In the standard version of the activity, the computer provides the child with a target number that the child must place on a number line. The child must click on the number line where he thinks the target number belongs. Once the child clicks, an arrow appears. The child can move the arrow to adjust his answer using the arrow keys or clicking in a different spot on the line. Once the child believes he has found the target number, he presses the space bar to submit his answer and receive feedback.

To scaffold a specific strategy, the arrow can already be presented to the child in a specified location. The arrow serves as a visual cue to help the child focus his attention on the relevant portion of the number line. Then, the child may move the arrow left or right to find the target number and submit his answer using the spacebar. For example, if a child must place 5 on a 0 to 100 line, the arrow will appear close to the left endpoint whereas if a child must place 80, the arrow will appear close to the right endpoint. Similarly, if a child must place 53, the arrow may appear close to 50, making it easier to use the midpoint strategy and encouraging the use of the most appropriate strategy based on the target number.

Conclusion Learning strategies is at least as important as memorizing facts or learning procedures. Cognitive psychology has identified the major strategies children use as well as the conditions under which the strategies can be effective. Software designers should aim to promote those strategies using the special affordances of computers, including visual representation, tools, setting problem parameters, feedback, and scaffolding.

Identify and Eliminate Bugs and Other Misconceptions

Children (and adults) make mistakes, some of which are unsystematic. Children may guess an answer if they don't understand the question, or if they don't know how to solve a problem. Unsystematic errors could also result from boredom or carelessness. On the other hand, children's mistakes may result from systematic use of an inefficient strategy or rule—a "bug"—that may have sensible origins (Ginsburg 1989). Computer programs are not the only operating systems that may suffer from bugs! For example, Sarah may say that $12 + 35$ is 38, $23 + 13$ is 54, and $1 + 25$ is 17. At first glance, the child's answers may seem to be absurd and nonsensical, and one may speculate that the child is simply guessing the answers. However, a closer look clarifies that Sarah is systematically using an incorrect rule to find the answers. She first adds the digits of the double-digit numbers, and then puts results together to form a new number: $12 + 35 = (1 + 2) + (3 + 5) = 38$.

When Bob was asked to identify the hen with more eggs, (see Fig. 9), he always responded with the hen on the right side. Was he guessing or did he simply like the hen on the right side (Fancy Pants) more than the one on the left side (Fluffy)? Or, is it possible that his mistake is based on an incorrect strategy use? We asked Bob to show us how he knows Fancy Pants (who had 4 eggs) has more eggs than Fluffy (who had 5). Starting on the left side, he counted all the eggs for both hens together, and said "Fluffy has 5, and Fancy Pants has 9, so Fancy Pants has more." Because systematic errors are fundamentally different from random mistakes, designers and educators must examine not only accuracy but also attempt to discover *why* the child was incorrect.

To create useful software, designers need to draw upon cognitive research to develop clear answers to the following three questions related to errors and misconceptions:

1. What are the possible bugs and other misconceptions underlying mistakes?
2. How do we assess the bugs, misconceptions and other sources of errors?
3. What can be done to help children eliminate the errors?

Possible Bugs and Their Sources The cognitive psychology literature (Brown and VanLehn 1982) provides detailed information concerning children's "buggy" solutions in arithmetic. For example, given the problem

$$\begin{array}{r} 21 \\ \underline{-19} \end{array}$$

children often get the answer 18, because they always "take away" the smaller number from the larger. Or given the problem

$$\begin{array}{r} 21 \\ \underline{+19} \end{array}$$

they get the answer 310 because they fail to regroup the ten (or "carry").

In our own work with *MathemAntics*, we found many systematic errors and misconceptions when children compare quantities of two sets. Some of these systematic errors were already described in the literature as far back as Piaget (1952a). Some children had no difficulty identifying the set with more objects when the spatial attributes and numerical magnitude were congruent (as when both the length and number of objects in one row is greater than the length and number of objects in the other row), but failed to answer correctly when the spatial and numerical properties of the sets were incongruent (as when the shorter row has more objects than the longer). These children were inappropriately using a visual comparison strategy.

What are the sources of errors? In many cases it is hard to choose among several possibilities. Some of the errors and misconceptions are directly related to the child's mathematical understanding. Saxe et al. (2010) found that although the number line affords representing powerful mathematical ideas, and many children have used the number line in the classroom, they have misconceptions about its key principles. When asked to place various targets on the line (e.g., 9, 12, 13) children placed the targets in the correct numerical order from left to right, but failed to space them correctly according to linear units. Our informal pilot work revealed similar findings. Second-grade children placed all targets (1–9) on an open number line marked 0–10 equidistant in relation to one endpoint while ignoring the other endpoint, resulting in a line clustered close to the left endpoint. These children reported that the number lines they constructed were correct, indicating they were attending to the order of numbers on the number line while ignoring the proportionality. In this case, the misconception stems from lack of understanding of the key features of the number line. Through a carefully designed tutorial integrating the number line with additive numerical units, Saxe et al. (2010) were able to mitigate these misconceptions.

However, other errors may be due to limitations in procedural and utilization skills, or components of executive functioning such as working memory, inhibition, task-switching, and updating (Bull and Scerif 2001) rather than a lack of understanding the mathematical concepts. It is important to consider domain general cognitive skills of a child and how these skills contribute to the child's understanding (or misunderstanding) of a topic. For example, a child whose comparison judgment does not match his counting may have forgotten the counting results by the time he judges which set has more (hence, limitation in working memory). Another explanation is that the child does not know the meaning of all the counting words (hence has not mastered the counting principles), or the child simply does not understand the question that has been asked.

To highlight another example, when completing the *MathemAntics* version of the number line estimation task, one first-grader worked with a number line marked from 0–10 on his first few sessions. Once he mastered the 0–10 line, he moved on to work on a number line marked from 0–100. When presented with 10 as a target number, he immediately placed it at the right endpoint. While he correctly identified the right endpoint as being 100 he continued to make a perseveration error: he was unable to inhibit the response of placing 10 at the right end point. This seems to indicate difficulty with inhibition and task switching as much as difficulty with the mathematics itself.

Identifying and Assessing Errors

In developing *MathemAntics*, we used careful observation to identify the bugs, misconceptions, and other sources of children's errors. We also conducted informal "clinical interviews" with the children as they used the software. We asked questions like, "How did you get that answer?" or "How did you know? Show me." In a later section, we expand on the rationale for use of the clinical interview.

We recorded our observations and interviews with *Silverback*© software (Clearleft Limited 2010), which keeps a video record of everything the child does on the computer screen and also makes a video of the child's face and anything said during the session. At the end, *Silverback*© exports a video showing both the child's work on the screen as well as her behavior, including her responses to clinical interview questions. This is an extremely efficient tool for recording, identifying, and later coding children's strategies and concepts as they use the software.

Having used observations and the clinical interview to obtain useful information about children's thinking, we are able to design "stealth assessment" of the child's learning. For example, if we learn that children typically use certain bugs, such as always subtract the smaller number from the larger, we can have the computer identify the wrong answers as those likely to have been produced by that bug: thus, $12 - 5 = 13$ and $12 - 3 = 11$. This method originates in the seminal work of Brown and Burton (1978).

To investigate whether a child uses area rather than number to compare the quantity of two sets, the software could analyze the child's answers to the problems in which area and number magnitudes are congruent and in those that are not congruent. If the child is always correct for the congruent problems, and not correct for the incongruent problems, the child is using a visual comparison strategy inappropriately. Subsequently, appropriate feedback and scaffolding may be provided to eliminative such systematic inappropriate strategy use. If we fail to conduct this kind of analysis, we might incorrectly conclude that the child's performance is simply a result of random response.

Helping Children Overcome the Errors Having identified children's systematically incorrect thinking (like buggy strategies), well-designed software needs to help children to understand the error of their ways (and the ways of their errors). Direct scaffolding is one approach. The software can first identify the child who fails to choose the hen with more eggs when area and numerical magnitudes are incongruent. Next the software can tell the child that he needs to count each set separately and then use this information to identify the set with more objects. Scaffolding with explanation and clarification is an even better approach. For example the software could arrange the eggs in rows and columns, match the eggs from one hen to the other, and point to the hen that has more eggs. Then, it could explain that greater area does not necessarily mean the higher number, and to know which hen has more, counting is a better strategy.

In some other scenarios, there is even a better approach, one that involves active participation of the child. The boy who failed to inhibit the response to place 10 at the right endpoint of a hundreds number line inspired the design of a "count by 10s" scaffold. The *MathemAntics* software can identify a child who is making this type of perseveration error (e.g. placing 2 in the place of 20), show him a number line with the decades marked, and then instruct him to count by 10s, as each decade number highlights along with an audio that speaks the decades name.

The simple theory underlying our approach is that you need to understand children's errors before you can help children to overcome them. And if the software can identify the thinking underlying the errors, then we can program the software to help the child construct a better understanding that will in turn eliminate the errors.

Conclusion Cognitive psychology can provide detailed information concerning children's systematic errors (or "bugs") in arithmetic. Guided by the cognitive principle of starting with the child's current state, designers need to produce software that can identify the errors and then correct them. Furthermore, designers should use clinical interviews in usability testing to identify systemic errors connected with the software itself. The basic principle is to identify, understand, and respond to the systemic ways in which users misunderstand the mathematics or the software.

Design Appropriate Physical Interactions

Until now, we have introduced each principle with important lessons from cognitive psychology and have stressed how to apply them to designing educational software. In the case of interactions, we must first begin with the possibilities that the technology affords. With computer software, a child may click, drag, or use the keyboard to interact with the program. As new technologies emerge, especially multi-touch devices, the ways in which the child can interact with the program evolve as well. Designers can now deploy a plethora of interface interactions such as touch, drag, swipe, squish, expand, rotate, blow, tilt, or shake, fling, and flick all of which can add to the interest, functionality, effectiveness, and creativity of the software.

Decisions about which interface interactions to include in the design are not trivial. Children are physically and cognitively different from adults and these differences have important implications for software design. Some design decisions are related to physical usability. Can the young child touch and drag an object on the screen? Can the child use her fingers to pinch objects so as to reduce their size? Other design decisions must be based on cognitive psychology. We focus here on how cognitive psychology can inform the following question regarding interface interactions: What are the cognitive benefits and limitations of different kinds of interface interactions, particularly on touch screen devices?

Benefits and Limitations Clearly, new and more complex interface interactions may help make the software more engaging and motivating, but can children use and learn from them? Shuler (2009) warns that the limiting physical attributes of mobile devices (such as restricted text entry and small screen size) as well as the usability difficulties of poorly designed interface interactions may distract children from their learning goals. Poorly designed applications fail to consider usability issues and motor development of children at various ages.

Young children may have physical difficulty with some standard (not touch-screen) computer interfaces. Using the mouse to click on an icon, drag items on a screen, rotate, tilt, and draw requires a degree of fine motor development typically lacking in young children. Hourcade et al. (2004) found that preschool children were slower and less accurate than older children in using the mouse for point-and-click tasks, especially when the target size was small. This difficulty can be attributed to under-development of fine-motor skills and is likely to distract from learning. Fisher (2012) recommends providing large icons with “large hotspots” around the target, and avoiding drag-and-drop if possible. Further, in the case of touch-screen design, she recommends leaving a “safe-zone” around the edges to avoid any accidental touch when holding the device, and having the software respond to the child’s touch only when an object is meant to be touched. The object then may make a noise, change color, wiggle, or make something happen in the activity.

Physical activity is important for designers not only because it can become a limiting factor for usability but also because it can promote the development of thinking. In Piaget’s (1952b) view, children’s cognitive development begins with physical activities like sorting objects, causing things to happen (like making an object move), comparing objects’ size and weight, and pointing to things during counting. Gradually these physical activities become internalized as cognition. Now the child can think about sorting, about causality, and comparisons. Now the child can do mental counting and addition.

Complex cognitive processes appear to be grounded in the body’s interactions with the learning environment (Wilson 2002). Multi-touch devices afford gestural interactions that build upon children's natural inclinations. Fingers are a child's first tools in learning mathematics. Children use their fingers to represent quantities, they point to objects to count them, and they use their fingers to add and subtract numbers (Butterworth 1999). A substantial body of literature calls attention to the significance of gesturing in learning and development (Dick et al. 2012; Goldin-Meadow and Alibali 2013; Ping and Goldin-Meadow 2008), and especially in learning and explaining mathematics (Cook et al. 2012; Goldin-Meadow et al. 2009). Gesturing might encourage children to extract meaning implicit in their hand movements (Goldin-Meadow et al. 2009); gesturing might help children to structure abstract ideas and map them metaphorically onto the visual space, the screen with which they are working (Kessel and Tversky 2006); gesturing might help children to focus their attention and embody the relevant information in the task.

Note that gesturing and hand movements matter, but also the form of gesturing is important in changing thought. In one study, Jamalian and Tversky (2012), showed that circular gesturing while explaining a cyclic sequence of events (such as the growth from seed to flower) helps adults to overcome their linear bias and instead to adopt more complex cyclical thinking. In another study, Segal (2011) investigated children's performance in discrete (single-digit addition) and continuous (number line estimation) mathematics problems while controlling for the type of gesture. In the continuous mathematics problem condition, 1st and 2nd grade children were asked to estimate a quantity on a number line shown on a computer interface. In this condition, children became more accurate when they dragged their finger or mouse on the number line as opposed to when they simply pointed and clicked on the number line. The gesture and the nature of the problem are both continuous. On the other hand, when solving addition problems, children were more accurate and adopted more advanced strategies when they pointed or clicked on each object to be counted/added comparing to clicking on a single button representing the total number of objects to be added. Segal argues that performance improves when the form of gesture is congruent with the problem-solving thinking process, and worsens when gesture is incongruent.

The educator's job is to help the child progress from physical to mental activity. Work with physical and virtual manipulatives should be designed to promote thinking, not merely playing with physical and virtual objects, no matter how attractive they may be. Engagement with physical and virtual objects is educationally useful only insofar as it eliminates the need for itself because what was on the "plane of action" is now transformed in the "plane of thought."

Given all this, a fundamental cognitive design principle is that the type and nature of gestures and other physical movements on both touch and mouse dominated screens can have an important influence on children's strategies and accuracy. This principle guided our design of the *MathemAntics* version of the number line activity. To allow flexible strategy use, we included many ways the child can place targets on the line. The child may use the mouse to click anywhere on the line, or use the left and right arrow keys to make small adjustments. Usability and play testing revealed that some children used the arrow keys to count discrete units from zero to reach the target number. Such counting strategies are useful, but inefficient and time-consuming, especially for larger target numbers (such as 98 on a 1–100 number line). To discourage the use of this strategy, we revised the activity so that the arrow keys still make small adjustments, but these adjustments no longer correspond to correct discrete units. Through feedback, the child will quickly discover that using the arrow keys to count to the target number will not lead to the correct answer and will encourage strategies based on approximate reasoning.

When implementing interactions, designers must also consider how the child will indicate he is ready to submit his answer. Far too many applications allow the child to succeed "by accident." For example, one simple enumeration app asks the child to put out 4 tokens, and the child may begin dragging tokens. But a close look at the child's strategy raises the possibility that he may be unsystematically dragging as many tokens as he can without counting or may be counting incorrectly. Yet once the child puts out 4 tokens, the trial ends and the child is given positive feedback about his performance, despite the fact that the child may have been trying to put out more than 4. This child does not have the option to carry out his plans and thus is missing an important learning opportunity that might result from his failure. We considered this issue in the design of *MathemAntics* and included a way for the child to indicate he is done working and ready to submit his answer. In our number line activities, the child has the flexibility of moving the cursor along the number line until he chooses to press the spacebar, which submits the answer and then results in the appropriate feedback.

Conclusion Computers allow children to interact with the computer interface in many different ways, from clicking and dragging to pinching with the fingers. Designers need to pay close attention to usability issues resulting from young children's limited fine motor skills. But more importantly, cognitive psychology suggests that children's gestures and other movements have important influences on what they learn and how they think, and that computer software needs to engage children in meaningful gestural interactions that promote deep learning.

Integrate Narratives and Stories with the Mathematical Concepts

Cognitive development in general and development of mathematical concepts in particular should take place in the context of supporting environments (Gelman et al. 1991). These environments can be social, cultural, natural (Gelman et al. 1991), or virtual. Gelman and Gallistel (1986) highlight the importance of meaningful context for measuring and promoting young children's number competence. Carpenter and Moser (1984) show that children's solutions to addition and subtraction problems reflect the semantic structure of the problem. Carpenter and Moser (1984) report that different contexts for subtraction word problems such as separating, joining, part-part-whole, comparison, influence the child's strategy use and responses. Other researchers (Barlow and Harmon 2012) point out benefits of problem contexts such as balancing and jig-saw problems for thinking about equality and the meaning of the equal sign.

Well-designed software can make learning meaningful through use of narratives. Narratives can help children to elaborate their everyday mathematics, for example by considering the number of monkeys falling off a bed one at a time. Narratives can help children synthesize concepts from the ordinary world with those in the abstract world of formal mathematics, for example by mathematizing the unfortunate monkeys as a series of subtraction statements. Narratives can give meaning to otherwise hard-to-comprehend symbols and operations, helping children to appreciate mathematics as a problem-solving discipline as opposed to a collection of meaningless facts and procedures to be memorized. In order to be successful, the goals of the narrative should be congruent with the goals of the mathematics. An effective narrative will assist the child in focusing attention on the relevant mathematical information, rather than distracting from it (Lepper and Malone 1987).

Stories, narratives, and fantasy also have the potential to enhance the learning experience by increasing motivation and engaging the learner (Malone and Lepper 1987). A meaningful narrative may help a child identify with the story and connect with the characters on an emotional level. Thus, the child's interest in the activity grows which in turn may build intrinsic motivation—interest in the activity for its own sake.

As promising as narratives are for learning, some designers fail to use them effectively. All too often the narrative is violent frosting on the number fact cake, as when the child is asked to shoot down correct number facts floating in the mathematical sky. Sometimes developers use a narrative that does not support or interferes with a meaningful learning process, as in the case of Rocket Math (Russell-Pinson 2012). In this game, students build a rocket and launch their rocket into space for math-based missions like solving addition or multiplication equations. Students earn "money" for their successful completed missions and they can use that money to upgrade and improve their rockets. The game is fun to play for many children, but in our own testing, we realized that students often spend more time buying boosters, fins, and decorations for their rocket than solving the math problems. Some children have even difficulties in building a rocket in the first place to launch into space so their rockets cannot even fly to solve the missions; for these children, the game-play gets focused on "how to build a good rocket" rather than learning the math facts.

We propose that it is important to consider the following three questions regarding to stories and contexts when designing learning activities.

1. What are the different contexts in which to situate an abstract concept?
2. What does the context add to the learning experience?
3. How could we integrate stories and the mathematical concept?

We illustrate the role of narratives and context with examples of *MathemAntics* activities focused on numerical relations and the equivalence of two sets. By definition, two sets are numerically equivalent if for every object in one set there is one and only one object in the other set.

Recognizing numerical equivalence between two sets, despite any visual differences, is a fundamental numerical competence (Mix 1999) that many adults may take for granted, but young children find cognitively challenging. Narratives can

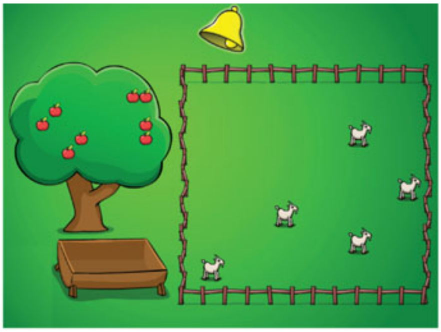
Fig. 13 Bedtime activity

situate this abstract concept in a meaningful context, helping children to better comprehend it. Containers, for example, are a natural match for objects. Children could start thinking about one-to-one correspondence by placing objects in containers and checking whether all objects have a container to go in to. In *MathemAntics*, we have chicks and beds, birds and nests, apples and baskets (see Fig. 13). Children are asked to put each chick in a bed, each bird in a nest, each apple in a basket, and check whether there are the same in number.

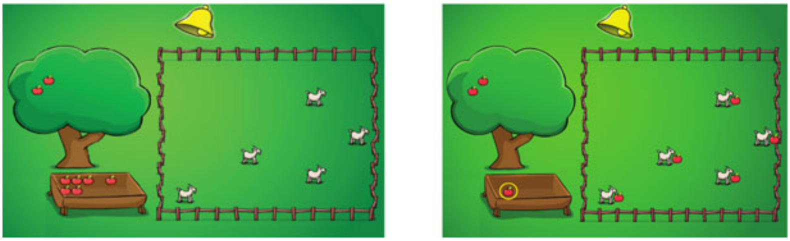
Feeding animals is another natural context in which the concept of numerical equivalence could be situated. Children are asked to pick as many apples as there are goats to feed, and ring the bell when they are done. After ringing the bell, apples get distributed among the goats and the child can check if he picked more, fewer, or just the same number of apples as there are goats (see Fig. 14).

In another activity we use the social schema of "ownership" and "having" to situate the concept of equivalence. Remember Fluffy and Fancy Pants? In earlier levels of the activity we ask children to judge whether the two hens *have* the same number of eggs. In more advanced levels of the activity, children are asked to make Fluffy and Fancy Pants have the same number of eggs, or make one of them have more or fewer eggs than the other. Children can touch the chicken's belly to lay an egg or touch an egg to hatch it.

"Friendship" in the sense of pairing is another everyday social schema that can bring up the idea of equivalence. In the *Matcher* activity children are in charge of finding friends for the chicks who are new to the farm (see Fig. 15). The child is asked to open the corral that has the same number of animals inside it as there are chicks so that each chick will have a friend. If the child opens the horses' corral, they come out, pair up with the chicks and dance in pairs. If the child opens the goats' corral, they come out, pair up with chicks, and the two extra goats weep complaining that they do not have any friends.



Task- Pick as many apples as there are goats

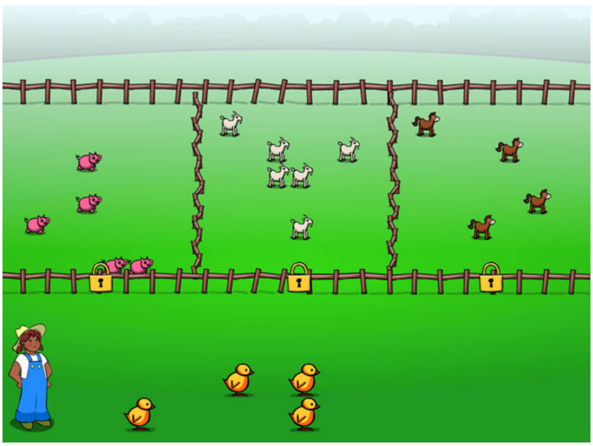


Before ringing the bell

Feedback shown after ringing the bell, highlighting the extra apple in the basket

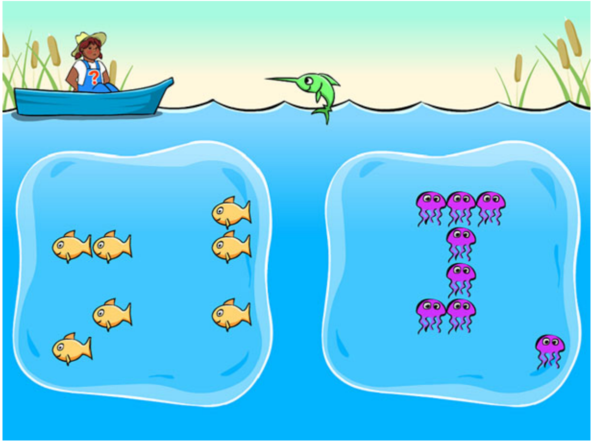
Fig. 14 Feeding Animals activity

Fig. 15 Match Friends activity



In more advanced activities we ask the child to directly compare the number of objects in two sets. Children are asked whether there is the same number of yellow fish as green fish, to pop the bubble that has more fish in it, or pop the bubble with fewer fish. (See Fig. 16. Note that the green swordfish pops the bubble.)

Fig. 16 Fish in the pond narrative in the Pop the Bubble activity



In all these activities, we introduce formal mathematical language such as "same", "more", "fewer", "most", "fewest", "greater", "less", verbal and written number words, mathematical symbols such as "$> = \neq <$", and number sentences. The formal mathematics is thus integrated into the overall narrative as part of instruction, feedback, scaffolding, or response mechanism.

Elements of narratives other than the content may also enhance the learning experience. For example, pedagogical agents could be characters in the story and play a central role in the learning process. In *MathemAntics*, we have a farmer who asks children to feed animals in the farm, count number of chicks in the barn, play with the fish in the pond, make the hens lay equal numbers of eggs, or estimate the amount of dough required for making cookies. The farmer provides feedback and scaffolding and rewards children with tools such as the "line-up" tool, or the "pair-up" tool, which children can use to solve more challenging problems.

Conclusion For learning to be effective, it may benefit from being situated in a meaningful context that gives it a purpose. Stories can promote and motivate mathematics learning. Unfortunately, some software employs narratives that are essentially irrelevant to the mathematics and also entail unnecessary violence, like shooting down rockets. But narratives and characters can promote meaningful mathematics learning if they are congruent with the mathematical ideas to be learned.

Evaluations and Assessments

Cognitive principles can guide evaluation and usability testing of software designs, and can also provide useful approaches to formative and summative assessment and to the evaluation of achievement and software effectiveness.

Cognitive Evaluation and Usability Testing of Software

At the beginning, designers should determine whether the software under development is congruent with the cognitive design principles described above. For example, does the software deal with the appropriate cognitive content? Does it promote thinking as well as procedure and memorized facts? Does it employ powerful visual models? Does it offer powerful tools? Does it identify bugs and misconceptions underlying errors? We propose that designers not conversant with the cognitive psychology of children's thinking and learning cannot create educationally effective software. The lack of this knowledge is one reason why many apps and software programs are of poor quality. What to do? One solution is to collaborate with psychologists. Every design company should hire at least one person knowledgeable in cognitive psychology!

Even if the software seems to pass the cognitive psychology design test, designers should conduct usability testing to see whether the software "works" in several senses. Designers want to determine whether the software contains any bugs, whether the children understand the function of a particular feature (like an icon that must be pressed to submit an answer), whether children can navigate through the material effectively (first use this icon and then use that), and whether they misinterpret the purpose of the activity. All good designers are well aware of the need for early and fast usability testing. The common practice is to closely observe the user interact with the software; users may ask questions and are encouraged to express their difficulties, feelings, or comments; however, designers are encouraged not to interfere with the experience and not to answer any of those questions. The goal is to investigate whether the user interacts with the software as predicted, if the user enjoys the interaction, and whether there are any usability or navigation problems.

Although observational usability testing can yield important information, we propose that observation is not sufficient and must be supplemented by intensive questioning—a "clinical interview usability method" that in turn leads to methods for helping children to understand the task and to perform more effectively.

The clinical interview method, originally developed by Piaget (1976), involves a flexible questioning of the subject, child or adult (Ginsburg 1997). The interviewer attempts to identify the thinking underlying the subject's overt behavior by asking relevant questions, modifying them, and following up with probing questions such as: "Why did you do that?," "How did you know?," and "How did you figure it out?" These kinds of questions need to be guided by as much knowledge of children's thinking as possible. For example, the interviewer who has extensive knowledge of the bugs that typically infest children's calculation methods is more likely to ask penetrating questions than the interviewer ignorant of them. In general, the more you already know about children's thinking (or almost anything else) the more can you learn about it. But you have to start somewhere, and the question "How did you do it?" is a good first step.

The challenge for designers becomes how to embed clinical interview methods into the usability testing of the software. Again, the main solution is to collaborate with cognitive psychologists. Every design company should hire at least one person knowledgeable in the clinical interview (as well as in cognitive psychology as described above—ideally the same person).

Formative and Summative Assessments

As we showed earlier, the computer software can be used to conduct "stealth assessment" of the child's learning. For example, if we learn from cognitive psychology or pilot work that children consistently believe that physically larger sets are more numerous than smaller ones (for example, 3 elephants are "more" than 5 ants), we can program the computer to identify these "size bugs" by tracking wrong answers in response to similar problems contrasting size and number. The software can also identify such important behaviors as accuracy; latency in response to different types of problems (some of which benefit from speed, like retrieval of number facts) and some of which, by contrast, require careful contemplation (as in the case of word problems); and the frequency of certain kinds of choices (as when the child is given the opportunity to employ different tools and strategies).

Stealth assessment can provide teachers with valuable "formative assessment" reports on individual students or the whole class, as well as suggestions of activities that might benefit individual students or even the whole class. For example, if a student can accurately retrieve "number facts" (like $2 + 3$) but has no strategies to figure out facts that have not been memorized, the computer can offer activities designed to promote use of strategies. By contrast, the student who possesses good strategies for figuring out number facts but cannot recall them quickly and accurately may benefit from drill. The suggestions need not be limited to computer activities, but could also involve textbook lessons, work with physical manipulatives, or even a mandate to avoid spending too much time working with computers and instead to go out and play in the non-virtual world of sun and fresh air.

Stealth assessment can also be extremely valuable for "summative" data by tracking progress during the school year and providing pre- and post-test indicators of student achievement. Unlike most conventional achievement tests, stealth assessments also have the ability to describe the trajectory of the individual child's (and the class of children's) learning. The assessments can depict the evolution of accuracy, concepts, and strategies, and even engagement and interest (Rodrigo and Baker 2011). Moreover, these portraits are based on many data points—literally thousands for children using the computer on a daily basis. The volume of data can overcome the inevitable noise produced by children's fluctuating attention and other sources of error variance. Moreover, the assessments draw upon problems of obvious educational relevance and thus have good "face validity." Conventional achievement tests may correlate highly with other conventional achievement tests, but this may be merely an artificial dance in which the blind lead the blind. What could be more "valid" than reports using very large amounts of data to depict the development of children's performance, strategies and concepts?

Finally, summative assessments of this type can serve as the foundation for evaluating particular software programs or apps. The summative data can be used to compare the overall effectiveness of different software programs, but more importantly can be used to compare the effectiveness of particular activities within a software program. For example, one activity within a comprehensive computer program may be successful at teaching concepts but not so effective at promoting recall, whereas an activity within another program may promote recall but not concepts. Assessments of this type can help a school district evaluate educational software in a more nuanced manner than can a conventional evaluation providing pre- and post-tests of dubious validity. And they can help designers improve specific activities within a software package.

Conclusion Cognitive principles can provide a preliminary basis for evaluating software. They can reveal whether the software as constituted seems to focus on important aspects of learning and use sound pedagogy. The cognitive principles and the method of clinical interview can also guide usability research. Sound software based on these principles can provide teachers with formative assessments of student performance, learning and thinking, and suggest approaches to instruction. The software's summative assessments can serve as valid indicators of student achievement, and might even eliminate the need for traditional evaluation measures or at least assign them a secondary role.

From Software to Research

We have shown that cognitive psychology can and should guide software design, evaluation and assessment. Here we argue that high-quality software offers new ways to conduct cognitive research and a powerful lens for examining the child's mind.

Most research studies on the development of mathematical thinking use a cross-sectional approach to investigate children's current cognitive status at several age levels. The outcomes of this kind of research have been enormously useful for both psychology and education, and have indeed revolutionized our views of children's mathematical development. We know a good deal, for example, about young children's everyday addition and how it influences the understanding of the symbolic addition taught in school. The research is not only of theoretical interest but can be used to guide teaching and assessment.

At the same time, this important research is limited in a key respect. Vygotsky (1978) makes a distinction between the "actual developmental level" (p. 85), which is the child's current state, vs. the level of "potential development as determined through problem solving under adult guidance or in collaboration with more capable peers" (p. 86). The essence of the distinction is the idea that the child's current performance may not indicate underlying competence and potential for learning.

Vygotsky stresses the importance of examining the "...dynamic mental state, allowing not only for what has been achieved developmentally but also for what is in the course of maturing" (p. 87).

We think it is fair to say that most research studies on cognition generally, and on the trajectories of mathematical development in particular, do not follow Vygotsky's advice. Instead the research often uses cross-sectional studies to investigate children's current cognitive status at several age levels. Because this enormously valuable research largely does not deal with children's "dynamic mental state," we need to take the idea of "developmental trajectories" with a dash of Vygotskian salt. It is possible that current views of developmental trajectories underestimate what children can do and learn.

One way to investigate this issue is by closely examining children's mathematical thinking in the context of stimulating software such as *MathemAntics*. As noted earlier, Papert (1980) made the argument that children's mathematical thinking may look very different in rich "microworlds"—artificially designed "mathemagenic" environments—than it would in the context of the not-very-dynamic school or home. This argument is especially important in the case of disadvantaged children and those with disabilities. Our own preliminary observations suggest that low-SES children may perform much better than expected when engaged in *MathemAntics* software. And some early research on LOGO suggests that mathemagenic computer environments enable severely disabled individuals to perform at unexpected levels (Weir et al. 1982).

Not only does the mathemagenic environment provide the context for dynamic learning, but the software also supplies the means for studying it. *MathemAntics* includes a control panel, hidden from the child, that allows the adult to vary key problem features, like the number, size and shape of virtual objects, the availability of tools, and the nature and timing of feedback. Other software (unless it is extremely simple and rigid) must have controls like these as well, even if they are hidden from the user and anyone else. The control panel is an experimenter's delight. It allows for easy manipulation of all or any of the available variables. In effect, the panel provides the opportunity for factorial designs. (We plan to make our *MathemAntics* software and panel available to researchers willing to share data with us.) And because children do the activities repeatedly, over time, the software provides the opportunity for microgenetic research—the investigation of learning over a not too brief time period of say a few weeks or months. Of course, if children are engaged in the various activities over a period of years, the opportunities for longitudinal research are evident.

Further, as we have seen, the software can help to measure children's performance, accuracy, latencies, and even the kinds of strategies they employ. This stealth assessment can be useful for the researcher, as well as the teacher. But in either event, the development of stealth assessment requires considerable pilot research. Some may require close observation of children working with the software, as we described earlier. Of course, stealth assessment may not produce a complete picture, and therefore may need to be supplemented by other kinds of measures (like coding video segments of children as they work with computers). But the software does make possible computer generated descriptive reports, like frequencies of strategy use, or the conditional probability of strategy use given certain conditions. For example, does the child use the counting on strategy for addition on the problems for which it is most appropriate? The reports can be imported into a statistical program so as to permit instant analysis of very large amounts of data. In principle, the software could easily generate reports and analyses of thousands (why not millions?) of children working with the software.

Conclusion The development of exciting, meaningful and motivating software provides researchers with the opportunity to study children's learning in a context that may elicit from them more advanced performance and thinking than they display in standard experimental tasks. The software may reveal surprising deviations from known developmental trajectories, especially in disadvantaged and handicapped children. In any event, the software makes it feasible to conduct microgenetic and longitudinal studies, to gather data on very large numbers of children, and conduct stealth assessments that reveal much more than accuracy. These possibilities may in turn require the development of new statistical approaches to apply to massive microgenetic research studies.

Final Remarks

We have argued that cognitive principles can and should shape the development of software, which in turn can improve learning, teaching and testing. We used examples from *MathemAntics* to show that such software is not a figment of our imagination. But examples—even though providing a kind of existence theorem—are insufficient in the absence of solid data. Skepticism is appropriate because many educational innovations fail, and computers are no exception to the rule.

But suspend disbelief and suppose that we can create an effective and comprehensive system of early mathematics software guided by cognitive principles. The possibilities for transforming education are enormous.

The software will enable children, especially the disadvantaged, to reach higher levels of mathematics achievement than they do now. They will learn both concepts and skills; they will think as well as remember; and they will use sensible strategies as well as apply standard algorithms. And they will enjoy learning meaningful mathematics.

The software will enable children to explore and learn on their own and at their own pace. This can be a lifeline for students with weak teachers—who are unfortunately more numerous than we would like. The software will also help children work with each other on common problems, engage in productive argument, and share solutions.

Children will not make a strong distinction between literature and mathematics. They will read and engage with mathematical stories and enjoy both the mathematics and the stories.

Comprehensive mathematics education software will supplement, eliminate, or transform traditional textbooks. Static two-dimensional representations on paper are clearly ineffective for portraying many kinds of transformations (like subtraction) and other mathematical processes (like splitting a segment on a number line into a thousand pieces) and for providing opportunities for interactive learning. At the very least, comprehensive mathematics software can supplement traditional textbooks. But two other possibilities are more attractive. One is replacing textbooks entirely, especially for young children who cannot read well. Older children might be better served by a hybrid textbook/software system that preserves the written word but also offers interactive software. In other words, textbooks can take the form of e-books that embed the mathematics software in a coherent manner.

As children in the class work on computers, the teacher can attend to the needs of individual children more frequently and effectively than is possible in a large classroom without computers.

The software-based formative assessments will help teachers learn to understand children as individuals and will guide teaching. The assessments can be seen as a kind of professional development for teachers who know little about underlying principles of student learning and thinking and who do not know how to conduct effective assessments of individual students.

The assessments can change the practice of evaluation. Because it provides so much rich data, stealth assessment should largely replace other assessment and testing procedures. What could be more ecologically valid and useful than detailed portraits of children's learning over the course of a year or longer?

Software can set the stage for a different kind of basic research that examines children's learning and thinking in rich and adaptive educational environments over long periods of time. The number of children and number of "trials" will be enormous—for example, hundreds or even thousands of children working with computers all year long and providing many thousands of data points. In this kind of adaptive learning environment, no two children may have exactly the same learning experiences or the same number of them. To deal with these complexities, use of novel statistical methods will be essential. Analyses created for basic factorial designs will not be helpful. Use of effect size to determine the significance of mean differences in achievement will become obsolete.

And finally we observe that apps can introduce a large amount of whimsical fun (think Fluffy and Fancy Pants) into children's mathematics learning. This is not trivial: the antic (as in *MathemAntics*) is intended not only to amuse but to show that thinking needs to be liberated from dull convention.

References

Alibali, M. W. (1999). How children change their minds: strategy change can be gradual or abrupt. *Developmental Psychology*, *35*, 127–145.

Barlow, A., & Harmon, Sh. E. (2012). Problem contexts for thinking about equality: an additional resource. *Childhood Education*, *88*(2), 96–101.

Baroody, A. J. (2004). The developmental bases for early childhood number and operations standards. In D. H. Clements, J. Sarama, & A.-M. DiBiase (Eds.), *Engaging young children in mathematics: standards for early childhood mathematics education* (pp. 173–219). Mahwah: Erlbaum.

Booth, J. L., & Siegler, R. S. (2006). Developmental and individual differences in pure numerical estimation. *Developmental Psychology*, *41*(6), 189–201.

Brown, J. S., & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, *2*, 155–192.

Brown, J. S., & VanLehn, K. (1982). Towards a generative theory of "bugs". In T. P. Carpenter, J. M. Moser, & T. A. Romberg (Eds.), *Addition and subtraction: a cognitive perspective* (pp. 117–135). Hillsdale: Erlbaum.

Brian, M. (2012). Apple: 1.5 million iPads are used in educational programs, with over 20,000 education apps [Web log post]. Retrieved January 19, 2012, from http://thenextweb.com/apple/2012/01/19/apple-1-5-million-ipads-in-use-in-educational-programs-offering-over-20000-education-apps/.

Bull, R., & Scerif, G. (2001). Executive functioning as a predictor of children's mathematics ability: inhibition, switching, and working memory. *Developmental Neuropsychology*, *19*(3), 273–293.

Butterworth, B. (1999). *The mathematical brain*. London: Macmillan.

Butterworth, B., & Laurillard, D. (2010). Low numeracy and dyscalculia: identification and intervention. *ZDM Mathematics Education*, *42*, 527–539.

Carpenter, T. P., & Moser, J. M. (1984). The acquisition of addition and subtraction concepts in grades one through three. *Journal for Research in Mathematics Education*, *15*, 179–202.

Chard, D. J., Baker, S. K., Clarke, B., Jung, J. K., Davis, K., & Smolkowski, K. (2008). Preventing early mathematics difficulties: the feasibility of a rigorous kindergarten mathematics curriculum. *Learning Disability Quarterly*, *31*, 11–20.

Chen, Z., & Klahr, D. (1999). All other things being equal: children's acquisition of the control of variables strategy. *Child Development*, *70*, 1098–1120.

Clearleft Limited (2010). Silverback (version 2.0) [computer application software]. Available from. http://silverbackapp.com/.

Clements, D. H., & Sarama, J. (2007). Early childhood mathematics learning. In F. K. Lester Jr. (Ed.), *Second handbook of research on mathematics teaching and learning* (pp. 461–555). New York: Information Age.

Clements, D. H., & Sarama, J. (2008). Experimental evaluation of the effects of a research based preschool mathematics curriculum. *American Educational Research Journal*, *45*, 443–494.

Common Core State Standards Initiative (2010). Common Core State Standards for mathematics. Retrieved from http://www.corestandards.org/assets/CCSSI_Math%20Standards.pdf.

Cook, S. W., Yip, T., & Goldin-Meadow, S. (2012). Gestures, but not meaningless movements, lighten working memory load when explaining math. *Language and Cognitive Processes*, *27*(4), 594–610.

Cordes, S., & Gelman, R. (2005). The young numerical mind: when does it count? In J. Campbell (Ed.), *Handbook of mathematical cognition* (pp. 127–142). London: Psychology Press.

Cross, C. T., Woods, T. A., & Schweingruber, H. (Eds.) (2009). *Mathematics learning in early childhood: paths toward excellence and equity*. Washington: National Academy Press.

Cunha, F., Heckman, J., Lochner, L., & Masterov, D. (2006). Interpreting the evidence on life cycle skill formation. In E. Hanushek & F. Welch (Eds.), *Handbook of the economics of education* (pp. 307–451). North Holland: Elsevier.

Dewey, J. (1976). The child and the curriculum. In J. A. Boydston (Ed.), *John Dewey: the middle works, 1899–1924. Vol. 2: 1902–1903* (pp. 273–291). Carbondale: Southern Illinois University Press.

Diamond, A. (2008). Cognitive control (executive functions) in young children: relevance of what we know to what can be done to help children. In *Emotion regulation, children's brains and learning, plenary session presented at head start's ninth national research conference*, Washington.

Dick, A., Goldin-Meadow, S., Solodkin, A., & Small, S. (2012). Gesture in the developing brain. *Developmental Science*, *15*(2), 165–180.

Duncan, G. J., Dowsett, C. J., Claessens, A., Magnuson, K., Huston, A. C., Klebanov, P., Pagani, L. S., Feinstein, L., Engel, M., Brooks-Gunn, J., Sexton, H., Duckworth, K., & Japel, C. (2007). School readiness and later achievement. *Developmental Psychology*, *43*(6), 1428–1446.

Feigenson, L., Dehaene, S., & Spelke, E. S. (2004). Core systems of number. *Trends in Cognitive Sciences*, *8*(10), 307–314.

Fisher, C. E. (2012). *Guidelines for successful mobile interactive apps for children*. Paper presented at the annual meeting of game developer conference, San Francisco, CA.

Fuson, K. C. (1992a). Research on learning and teaching addition and subtraction of whole numbers. In G. Leinhardt, R. T. Putnam, & R. A. Hattrup (Eds.), *The analysis of arithmetic for mathematics teaching* (pp. 53–187). Hillsdale: Erlbaum.

Fuson, K. C. (1992b). Research on whole number addition and subtraction. In D. Grouws (Ed.), *Handbook of research on mathematics teaching and learning* (pp. 243–275). New York: Macmillan.

Fuson, K. C., Richards, J., & Briars, D. J. (1982). The acquisition and elaboration of the number word sequences. In C. Brainerd (Ed.), *Progress in cognitive development: Vol. 1. Children's logical and mathematical cognition* (pp. 33–92). New York: Springer.

Gelman, R. (1993). A rational-constructivist account of early learning about numbers and objects. In D. Medin (Ed.), *Learning and motivation* (Vol. 30, pp. 61–96). New York: Academic Press.

Gelman, R., & Gallistel, C. R. (1978). *The child's understanding of number*. Cambridge: Harvard University Press.

Gelman, R., & Gallistel, C. R. (1986). *The child's understanding of number*. Cambridge: Harvard University Press.

Gelman, R., Massey, C., & McManus, M. (1991). Characterizing supporting environments for cognitive development: lessons from children in a museum. In J. M. Levine, L. B. Resnick, & S. D. Teasley (Eds.), *Perspectives on socially shared cognition* (pp. 226–256). Washington: American Psychological Association.

Ginsburg, H. P. (1989). *Children's arithmetic* (2nd ed.). Austin: Pro-Ed.

Ginsburg, H. P. (1997). *Entering the child's mind: the clinical interview in psychological research and practice*. New York: Cambridge University Press.

Goldin-Meadow, S., & Alibali, M. W. (2002). Looking at the hands through time: a microgenetic perspective on learning and instruction. In N. Granott & J. Parziale (Eds.), *Microdevelopment: transition processes in development and learning* (pp. 80–105). Cambridge: Cambridge University Press.

Goldin-Meadow, S., Cook, S. W., & Mitchell, Z. A. (2009). Gesturing gives children new ideas about math. *Psychological Science*, *20*(3), 267–272.

Goldin-Meadow, S., & Alibali, M. W. (2013). Gesture's role in learning and development. *Annual Review of Psychology*, *64*, 257–283. doi:10.1146/annurev-psych-113011-143802.

Granott, N. (2002). How microdevelopment creates macrodevelopment: reiterated sequences, backward transitions, and the zone of current development. In N. Granott & J. Parziale (Eds.), *Microdevelopment: transition processes in development and learning* (pp. 213–242). Cambridge: Cambridge University Press.

Hourcade, J. P., Bederson, B. B., Druin, A., & Guimbretiere, F. (2004). Differences in pointing task performance between preschool children and adults using mice. *ACM Transactions on Computer-Human Interaction*, *11*(4), 357–386.

Hoyles, C., & Noss, R. (2009). The technological mediation of mathematics and its learning. *Human Development*, *52*, 129–147.

Hoyles, C., Noss, R., & Adamson, R. (2002). Rethinking the microworld idea. *Journal of Educational Computing Research*, *27*(1), 29–53.

iTot Apps, LLC (2011). Toddler counting (version 1.5) [mobile application software]. Retrieved from http://itunes.apple.com/.

Jamalian, A., & Tversky, B. (2012). Gestures alter thinking about time. In N. Miyake, D. Peebles, & R. P. Cooper (Eds.), *Proceedings of the 34th annual conference of the Cognitive Science*

Society (pp. 551–557). Austin: Cognitive Science Society. IES Prize for Excellence in Research on Cognition and Student Learning (pp. 503–508).

Jordan, N. C., Kaplan, D., Oláh, L. N., & Locuniak, M. N. (2006). Number sense growth in kindergarten: a longitudinal investigation of children at risk for mathematics difficulties. *Child Development*, *77*(1), 153–175.

Kessell, A. M., & Tversky, B. (2006). Using gestures and diagrams to think and talk about insight problems. In R. Sun & N. Miyake (Eds.), *Proceedings of the 28th annual conference of the Cognitive Science Society* (p. 2528).

Kuhn, D., Garcia-Mila, M., Zohar, A., & Anderson, C. (1995). Strategies of knowledge acquisition. *Monographs of the Society for Research in Child Development* (Serial No. 245).

Laski, E. V., & Siegler, R. S. (2007). Is 27 a big number? Correlational and causal connections among numerical categorization, number line estimation, and numerical magnitude comparison. *Child Development*, *78*(6), 1723–1743.

Lepper, M. R., & Malone, Th. W. (1987). Intrinsic motivation and instructional effectiveness in computer-based education. In R. E. Snow & M. J. Farr (Eds.), *Aptitude, learning, and instruction: Vol. 3. Conative and affective process analyses* (pp. 255–286). Hillsdale: Erlbaum.

Malone, T. W., & Lepper, M. R. (1987). Making learning fun: a taxonomy of intrinsic motivations for learning. In R. E. Snow & M. J. Farr (Eds.), *Aptitude, learning, and instruction: Vol. 3. Cognitive and affective process analysis* (pp. 223–253). Hillsdale: Erlbaum.

Mclean, R. (2012). Space Math (version 1.4.2) [mobile application software]. Retrieved from http://itunes.apple.com/.

Mix, K. S. (1999). Similarity and numerical equivalence: appearances count. *Cognitive Development*, *14*, 269–297.

Mix, K. S. (2010). Spatial tools for mathematical thought. In K. S. Mix, L. B. Smith, & M. Gasser (Eds.), *The spatial foundations for language and cognition* (pp. 41–66). Oxford: Oxford University Press.

Moeller, K., Pixner, S., Kaufmann, L., & Nuerk, H.-C. (2008). Children's early mental number line: logarithmic or decomposed linear? *Journal of Experimental Child Psychology*, *103*(4), 503–515.

Moyer, P. S., Bolyard, J. J., & Spikell, M. A. (2002). What are virtual manipulatives? *Teaching Children Mathematics*, *8*(6), 372–377.

Moyer-Packenham, P. S., Salkind, G., & Bolyard, J. J. (2008). Virtual manipulatives used by K-8 teachers for mathematics instruction: considering mathematical, cognitive, and pedagogical fidelity. *Contemporary Issues in Technology and Teacher Education*, *8*(3), 202–218.

Moyer-Packenham, P. S., & Westenskow, A. (2013, in press). Effects of virtual manipulatives on student achievement and mathematics learning. *International Journal of Virtual and Personal Learning Environments*.

Mullis, I. V. S., Martin, M. O., Beaton, A. E., Gonzalez, E. J., Kelly, D. L., & Smith, T. A. (1997). Mathematics achievement in the primary school years. In *IEA's third international mathematics and science study (TIMSS)*. Chestnut Hill: Center for the Study of Testing, Evaluation, and Educational Policy, Boston College.

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

Petitto, A. L. (1990). Development of numberline and measurement concepts. *Cognition and Instruction*, *7*(1), 55–78.

Piaget, J. (1952a) *The child's conception of number*. London: Routledge & Kegan Paul Ltd. C. Gattegno & F. M. Hodgson, trans.

Piaget, J. (1952b). *The origins of intelligence in children*. New York: International Universities Press. M. Cook, trans.

Piaget, J. (1976) *The child's conception of the world*. Totowa: Littlefield, Adams & Co. J. Tomlinson & A. Tomlinson, trans.

Ping, R., & Goldin-Meadow, S. (2008). Hands in the air: using ungrounded iconic gestures to teach children conservation of quantity. *Developmental Psychology*, *44*(5), 1277–1287.

Ramani, G. B., & Siegler, R. S. (2008). Promoting broad and stable improvements in low-income children's numerical knowledge through playing number board games. *Child Development*, *79*(2), 375–394.

Rodrigo, M. M. T., & Baker, R. S. J. d. (2011). Comparing the incidence and persistence of learners' affect during interactions with different educational software packages. In R. A. Calvo & S. K. D'Mello (Eds.), *New perspectives on affect and learning technologies* (pp. 183–200).

Russell-Pinson, D. (2012). Rocket Math (version 1.8) [mobile application software]. Retrieved from http://itunes.apple.com/.

Sarama, J., & Clements, D. H. (2002). Linking research and software development. In *Research on technology in the teaching and learning of mathematics: cases and perspectives* (pp. 113–130).

Sarama, J., & Clements, D. H. (2009a). "Concrete" computer manipulatives in mathematics education. *Child Development Perspectives*, *3*(3), 145–150.

Sarama, J., & Clements, D. H. (2009b). *Early childhood mathematics education research: learning trajectories for young children*. New York: Routledge.

Sawyer, R. K. (2006). *The Cambridge handbook of the learning sciences*. *The new science of learning* (pp. 1–16). New York: Cambridge University Press.

Saxe, G. B., Earnest, D., Sitabkhan, Y., Halder, L. C., Lewis, K. E., & Zheng, Y. (2010). Supporting generative thinking about the integer number line in elementary mathematics. *Cognition and Instruction*, *28*(4), 433–474.

Schneider, M., & Siegler, R. S. (2010). Representations of the magnitudes of fractions. *Journal of Experimental Psychology. Human Perception and Performance*, *36*(5), 1227–1238.

Segal, A. (2011). *Do gestural interfaces promote thinking? Embodied interaction: congruent gestures and direct touch promote performance in math*. Unpublished Dissertation, Columbia University.

Shuler, C. (2009). *Pockets of potential: using mobile technologies to promote children's learning*. New York: The Joan Ganz Cooney Center at Sesame Workshop.

Siegler, R. S., & Booth, J. L. (2004). The development of numerical estimation in young children. *Child Development*, *75*(2), 428–444.

Siegler, R. S., & Jenkins, E. A. (1989). *How children discover new strategies*. Hillsdale: Erlbaum.

Siegler, R. S., & Svetina, M. (2006). What leads children to adopt new strategies? A microgenetic/cross sectional study of class inclusion. *Child Development*, *77*, 997–1015.

Starkey, P., & Klein, A. (2008). Sociocultural influences on young children's mathematical knowledge. In O. N. Seracho & B. Spodek (Eds.), *Contemporary perspectives on mathematics in early childhood education*. Charlotte: Information Age.

Vygotsky, L. S. (1978). *Mind in society: the development of higher psychological processes*. Cambridge: Harvard University Press.

Vygotsky, L. S. (1986). *Thought and language*. Cambridge: MIT Press. A. Kozulin, trans.

Weir, S., Russell, S. J., & Valente, J. A. (1982). Logo: an approach to educating disabled children. *BYTE*, *7*, 342–360.

Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin & Review*, *9*(4), 625–636.

Wilson, A. J., Dehaene, S., Pinel, P., Revkin, S. K., Cohen, L., & Cohen, D. (2006a). Principles underlying the design of "The Number Race", an adaptive computer game for remediation of dyscalculia. *Behavioral and Brain Functions*, *2*(1), 19.

Wilson, A. J., Revkin, S. K., Cohen, D., Cohen, L., & Dehaene, S. (2006b). An open trial assessment of "The Number Race", an adaptive computer game for remediation of dyscalculia. *Behavioral and Brain Functions*, *2*(20). doi:10.1186/1744-9081-2-20.