The purpose of this critical analysis was to investigate the important attributes of mathematical software designed for young children between the ages of four and seven and offer guidelines that will assist software designers in the production of developmentally appropriate educational software. In addition, teachers and parents may benefit from the information provided as they select educational software for their children. For the purpose of examination of educational software, the design criteria considered developmentally appropriate by researchers were combined with the design preferences of children.
According to the joint position statement provided by the National Council of Teachers of Mathematics (NCTM) and the National Association for the Education of Young Children (NAEYC), “high-quality, challenging, and accessible mathematics education for 3- to 6-year-old children is a vital foundation for future mathematics learning” (2002, p. 2) since children’s experiences during their early years influence their later performance in school as well as their life outside school. Educational organizations also support the use of computer technology in mathematics instruction as a means of supporting the goals of increasing both instructional opportunity and quality of the instructional experience (International Society for Technology in Education [ISTE], 2002; NAEYC/NCTM, 2002). Currently, computers and computer programs are being used with increased frequency in classrooms (Azevedo, 2005). With this increased use comes the challenge to provide young children with developmentally appropriate programs that meet their unique needs. This critical analysis attempts to inform designers of mathematics software for young children by examining the critical attributes of mathematical software intended for use by that population. Additionally, parents and teachers may find this analysis helpful as they endeavor to select appropriate educational software for their children. This analysis differs from other examinations because there was consideration of the literature on children’s own design preferences for educational software. Additionally, the analysis focuses specifically on mathematical software for young children.

**RESEARCHERS’ PERSPECTIVE**

Computer mediated learning has became more prevalent in day care centers, preschool programs, elementary schools, community centers, and after-school programs. As a result of the increased presence of technology in settings for young children there has been significant growth in new titles serving the early childhood educational market (Forgan & Weber, 2001; Haugland & Wright, 1997). Although there is a debate concerning the effects of computer use in teaching and learning, recent studies demonstrate that developmentally appropriate software can facilitate children’s learning, enable them to become more active participants in their own learning, and inspire their curiosity (Haugland & Ruiz, 2002; Haugland, Bailey, & Ruiz, 2002). In regard to the design criteria provided by researchers and educators, the following section will focus on identifying developmentally appropriate software in terms of its instructional content.
Defined Purpose of the Software

The purpose and content of a program is the first issue that should be considered in creating children’s software. The purpose of software should be clear to designers and instructional goals should be explained to learners to help them understand the purpose of the software and the relevance for engagement (Beale & Sharples, 2002; Buckleitner, 1999; Cottrell & Eisenberg, 1997; Downes, Arthur, & Beecher, 2001; Hall, Watkins, Davis, Belarbi, & Chandrashekara, 2001; Squires & Preece, 1999; The Cognition and Technology Group at Vanderbilt, 1997; Westrom, 2001). In addition, when creating software, designers should strive to create a learning activity that is both educationally beneficial and entertaining (Costabile, Angeli, Roselli, Lanzilotti, & Plantamura, 2003; Inkpen, 1997; Khalifa, Bloor, Middelton, & Jones, 2000; NAEYC, 1996). As noted in the NAEYC position statement (1996), it is important for children to play since children’s play influences their social, emotional, and cognitive development. Children can practice and enhance skills, learn to deal with emotions, and resolve conflicts while playing. Barta and Schaelling (1998) pointed out that games are rule-governed and allow engagement in estimation, prediction, and planning. Well organized and appropriate play, which is considered one of the six “universal mathematical activities” (Barta & Schaelling), should be an essential component of mathematics education software for young children.

Content

It is important for all children to develop a solid mathematical foundation during the early years of their lives, since mathematical understanding helps children make sense of the world around them (NAEYC, 1999; NCTM, 1995). NAEYC, in an effort to increase the quality of mathematics education for young children, has provided guidance to educators. First, consider and support children’s interest in mathematics and their disposition to use it to make sense of their physical and social worlds. Children’s engagement in normal daily activities, such as sorting or grouping, putting objects together or taking them away, counting, and identifying patterns, can be an opportunity to help children explore and use mathematics. Software designers should consider integrating these activities into educational software in order to enhance children’s interest in mathematics. Second, mathematics
understanding should be built on children’s experience and knowledge, including their family, linguistic, cultural, and community backgrounds. New experiences that are meaningful and connected with children’s prior knowledge and experiences help children develop their self-confidence, competence, and interest. Third, mathematics curriculum and teaching practices should be based on knowledge of children’s cognitive, linguistic, physical, and social-emotional development. Fourth, instructional practices should represent learners’ problem-solving and reasoning processes as well as representing, communicating, and connecting mathematical ideas, which are all considered by NCTM to be mathematical goals that students should attain. If we can presume that the same principles and instructional opportunities that should be present in mathematics classroom learning should be evident in educational software for mathematics then we can propose that software should be created for young children that helps them improve their problem solving, reasoning processes, representing, communicating, and connecting of mathematical ideas. Another recommendation is that there should be a bridge between mathematics and other authentic daily activities, such as sorting or grouping, counting, putting objects together or taking them away, and identifying patterns in order to take advantage of all instructional opportunities and provide children with a high quality math education. Children explore and make use mathematical operations in their daily activities. These kinds of activities can be used to introduce important mathematical ideas to children. For instance, designers could incorporate games with different scenarios where children may cook by following a step by step recipe that would enable children to count eggs or to measure milk. Or, children may manage money as they shop in a grocery store. Literacy, science, social studies, art, and music can be also combined with mathematics. One example of such a combination would be to add stories and songs with math themes that would provide a meaningful path for exploring and exchanging ideas about mathematical concepts. In addition, educators should actively introduce concepts, methods, and language through a range of appropriate experiences and teaching strategies. For instance, revisiting previously learned mathematical concepts provides greater opportunity for children to master the subject and thus forge stronger connections between the formerly learned concepts and new applications. Finally, children’s learning should be supported by thoughtfully and continually assessing their mathematical knowledge, skills, and employed strategies. Identifying learners’ strengths and needs, and adjusting instruction based on that information, is critical to helping children build mathematical competence and confidence. In brief, software developers should thoughtfully consider
the NAEYC recommendations for as they strive to improve the quality and
design of mathematical software for young children. Software designers
should also take into account the NCTM principals and standards for
mathematics. The principals and standards reflect basic precepts that are
fundamental to a high-quality mathematics education. As designers integrate
the recommended standards and specific expectations for mathematics
learning they will increase the opportunity for children to develop a solid
affective and cognitive foundation in mathematics.

Age, Gender, and Cultural Influences

The next issue to consider is the matching of individuals with software that
is age appropriate, culturally appropriate, and reflects gender equity.
According to the NAEYC Position Statements (1996), understanding—as it
relates to the following three categories—is valuable when designing
educational software for children:

1. knowledge of age-related human characteristics such as activities,
   materials, interactions, or experiences that will be safe, healthy,
   interesting, achievable, and also challenging to children;

2. knowledge of the strengths, interests, and needs of each child; and

3. knowledge of the social and cultural contexts in which children live to
   ensure that learning experiences are meaningful, relevant, and respectful
   for the participating children and their families (p. 6).

To elaborate on the NAEYC statements, first, the software in question
should be suitable for the learners’ age level so that they can find the
software interesting, and activities achievable and challenging; otherwise,
young children may get frustrated or bored. Second, while designers
certainly cannot know each child’s strengths, interests and needs, designers
can increase their understanding of children’s strengths, interests and needs,
and use that knowledge to provide opportunities for greater individualiza-
tion. Additionally, knowledge of gender related factors may help designers
create software that meets each gender’s strengths, interests and needs.
Finally, according to the NAEYC (1996), cultural and ethnic differences
should be considered as factors that influence children’s learning. In other
words, designers need knowledge of the social and cultural contexts in which children live in order to make new concepts and tasks meaningful, relevant, and respectful of the children. The following discussion further explores these important considerations.

**Age influences.** According to Rushton and Larkin (2001), “development occurs in a relatively orderly sequence, with later abilities, skills, and knowledge building on those already acquired” (p. 27). For instance, generally speaking, a child first learns how to sit, and then to crawl. With time, the child begins to walk. Briefly, learning occurs in a series of stages, which are predictable and stable (Case & Okamoto, 1996; Piaget, 1952). It is understood that there are certain times in an individual’s life when they are more receptive to some kinds of learning (Rushton & Larkin, 2001) and during these times, greater amounts of information can be processed by the brain (Wolfe & Brandt, 1998). For example, before children are taught data analysis and measurement, they should develop number sense and computational fluency since data analysis and measurement are strongly related to and sometimes dependent on these skills (Postlewait, Adams, & Shih, 2003). In addition, recent developments in brain research (Rushton & Larkin) emphasized the importance of developing and implementing learning environments that are appropriate for the learner’s particular developmental age. In brief, it is important for designers to create educational software built on developmentally appropriate learning objectives for the targeted age group.

All children have their own needs and interests; thus, the instructional format that the software uses should be adaptable to children’s developmental needs rather than requiring children to adapt to meet the demands of the software (DeLaurentiis, 1993; Haugland & Wright, 1997; Inkpen, 1997). Age appropriate environments would provide an atmosphere that would be appealing to the children a variety of activities for them (Silverstone, Whitehouse, Willis, McArdle, Jones, & O’Neill, 2001) and an opportunity for them to explore their feelings and ideas (Rushton & Larkin, 2001). Thus, software should meet children’s unique needs and do so in an appealing and child-friendly way.

**Gender influences.** In addition to age influences, gender differences also affect students’ learning. Sociological causes are one way to investigate gender differences and their influence on students’ academic performance (Davies, Marsh, & Millard, 2002; Körlin & Wrangsjö, 2001; Millard,
According to Millard, the strongest sociological influence is culture and the creation of a gender identity, characterized as “socially constructed maleness and femaleness” (p. 19). In other words, society’s expectations for males and females set up situations where individuals present and accomplish differently. For instance, in the past, girls were more frequently offered a wider range of writing and drawing materials, which were considered to be more of a feminine-type material, rather than mechanical toys or chemistry sets. In contrast, boys were expected to be more active than girls and participate more frequently in energetic and often competitive play, activities which required strength, power, and competitiveness (Davies et al., 2002; Martin & Hearne, 1989).

Mason, Kahle and Gardner (1991) argued that girls may be less likely to be socialized to build skills at home that are needed for science classes, so they participate in science activities less than boys in their early years. This stereotype can have a powerful influence on children’s thinking about their own competence and might limit their effort and persistence in activities considered age-appropriate. Hilton and Berglund (1974) stated that gender differences influence learners’ mathematical achievement because mathematics may be viewed as a “gender typed” area. If girls believe that mathematics is a male dominated area, they may be less willing to learn mathematics. To help girls and boys to understand they can be successful in science, mathematics, language, reading, and other skill domains, developers should be more equitable in terms of male and female characters in educational software and consider both girls’ and boys’ interest areas and interaction styles so they may better engage both genders’ attention to the software.

Gender differences may also affect students’ computer use. According to the NAEYC Position Statement (1999), while preschool-age girls and boys have equal interest in computers, over time girls spend less time in computer use than boys. Studies show that students’ attitudes toward computer use can be significantly different depending on gender (Comber, Colley, Hargreaves, & Dorn, 1997; Kirkpatrick & Cuban, 1998; Nelson & Cooper, 1997). For instance, Comber and colleagues found that boys are more likely than girls to effectively use computers and applications, such as word-processing, graphics, programming, and mathematics programs. Yelland (1994) found that boys were able to work faster and more efficiently than girls. However, after a while, girls performed better than boys. In addition, Sutton (1989) noted that girls lag behind boys in every category academically while they use computers. In brief, it might be concluded that gender is an important consideration for designers.
**Cultural and ethnic influences.** Cultural and ethnic differences should be considered as factors that influence students’ success. If a society becomes diverse, it is important for software developers to take into account diverse learners and individual differences in the design of software (Aldrich, 2002; Haugland & Shade, 1994; NAEYC, 1996, 1999; Shade, 1999). Nieto emphasized the connection between learning and culture and stated “...students of all backgrounds deserve the very best our society can give them, and ... their cultures, languages, and experiences need to be acknowledged, valued, and used as important sources of their education” (Nieto, 1996, p. xix). Thus, as the NAEYC position statement (1996) notes, developers should understand the influences of sociocultural context on learning and identify children’s developing competence. Specifically, in high quality mathematics education for three to six year-old children, mathematics experiences should be built on the learner’s family, linguistic, cultural, and community backgrounds (NAEYC/NCTM, 2002). Therefore, designers should consider cultural diversity in order to enable children to improve their mathematical understanding.

**Visual Environment**

Rusthon and Larkin (2001) have reviewed the literature about connecting developmentally appropriate practices to new understandings in the area of brain research. They found that there is a need for educators to provide children with increased opportunity to use “music, bodily-kinesthetic, visual-spatial, and interpersonal domains to learn and express understanding” (p. 31). Also, they noted that music, rhyme, and rhythm are directly connected with other basic functions of the brain, such as emotion, perception, memory, and even language. In addition to music, rhyme and rhythm, Rusthon and Larkin stated that pictures, symbols, and strong simple images have powerful influences on students’ learning. In other words, pictures, symbols, and images can help children keep information in memory for a long time. In brief, they concluded that art should hold an important place in the curriculum.

Colorful graphics and interesting animations keep children more focused on the subject. Wileman (1993) stressed the importance of graphics and stills and says children may encode and keep content in memory longer as well as retrieve it more easily through the use of well-designed visuals. Researchers

**Absence of Violent Images or Content**

Quality software should provide a safe environment that engages the learner in experiences that promote intellectual, physical, social, spiritual, emotional, and moral growth and development (Bushman & Anderson, 2002; Haugland & Wright, 1997; Heath, Bresolin, & Rinaldi, 1989; Huesmann & Skoric, 2003; NAEYC, 1994). However, today violence is often used in software and on web sites to attract children’s attention and, unfortunately, some children may find violent images or content entertaining. According to the NAEYC statement position (1994), children who are under seven or eight years old have great difficulty distinguishing “fantasy from reality and their ability to comprehend nuances of behavior, motivation, or moral complexity is limited” (p. 19); thus, children may think that violence is fun. In addition, the presence of violent images or content may cause children to become less sensitive to the pain and suffering of others; to become more fearful of the world around them; and to be more likely to act in aggressive or harmful ways toward others (NAEYC, 1994).

To emphasize the negative effects of violence, Haugland (n.d.) provided the following example:

What do we communicate to children when bombs are used to destroy pictures rather than an eraser? In one classroom, children quickly placed objects on the screen using a drawing program for the sole purpose of watching a bomb explode the objects on the screen. Children bombed the screen over and over. In software and web sites it is critical to recognize that children never experience the consequences of their violent behavior. They just quit and start over again as if they had never done the violent act. These communicates to children that violence is harmless. Yet in reality when something is bombed, it is destroyed. It is not possible to click a button and have objects and or people restored (Selecting Developmentally Appropriate Software, ¶20).
Also, Haugland and Wright (1997) stressed that software should not include any objects that young children could use to perform violent actions, such as guns, swords, bombs, or poison.

There are also several studies that sought to identify the effects on young children of playing an “aggressive” versus “nonaggressive” video game (Calvert & Tan, 1994; Irwin & Gross, 1995). Based on these studies, children who played an aggressive video game displayed more verbal and physical aggression, had a higher heart rate, reported more dizziness and nausea than did other children who did not play the aggressive video game. These studies did not measure children’s willingness to hurt anyone else; nonetheless, the results presented raise concerns about the potential impact of violent images or content on children’s mental and psychological development. This is certainly an issue that should be thoughtfully considered by designers of software for young children.

**Practice**

Multiple opportunities for practice are critical for young learners. Rushton and Larkin (2001) emphasized the importance of repetition and stated that “the child’s ability to learn and interpret new information is directly related to the frequency of prior experience with related ideas” (p. 30). They also pointed out that repetition is a critical factor for long term memory and argued that if children often repeat a new task, they can make more connections to what they already know. More specifically, according to the NAEYC/NCTM position statement (2002), students must learn mathematics with understanding by building new comprehension from experience and preceding knowledge, which can be done best through practice. To offer children opportunities to make a connection between a new subject and previously learned subjects, tasks or questions should be organized sequentially. After solving the first part of a task, users may be directed toward a conditional representation of the second part of the task (Baker & O’Neil, 2002). In addition, Baker and O’Neil suggested pointing out a conditional and corrected path in a problem. Based on the student’s answer, the software may provide either permission to continue or an opportunity to correct one’s answer. In other words, after users submit their response to the problem, they may have an opportunity to see whether their solution path is correct.
Assessment

Teachers spend as much as one third to one half of their time on assessment related activities (Stiggins & Conklin, 1992) in order to obtain information about students’ skills and potential, to improve instruction, and to be able to provide useful feedback for students (Baker & Mayer, 1999; Beevers, Goldfinch, & Pitcher, 2002; Gardner, 1991; NCTM, 1995; Salvia & Ysseldyke, 1995). Neuman, Copple, and Bredekamp (2000) stated the benefits of formative assessment by noting six primary purposes in education:

- monitor and document children’s progress over time;
- ensure that instruction is responsive and appropriately matched to what children are and are not able to do;
- customize instruction to meet individual children’s strength and needs;
- enable children to observe their own growth and development; and
- identify children who might benefit from more intensive levels of instruction, such as individual tutoring, or other interventions (p. 103).

More specifically, according to the NCTM (1995), while assessment may provide for these functions, it should also provide an opportunity for children “to formulate problems, reason mathematically, make connections among mathematical ideas, and communicate about mathematics” (p. 11). If designers apply this posit to software, then, applications should include assessments that infer students’ knowledge, understanding, reasoning, and thinking. While designed with older students in mind the Jasper Series (The Cognition and Technology Group at Vanderbilt, 1997), which provides multiple opportunities for problem solving, reasoning, communication, and making connections to other areas such as science, literature, and history. When used effectively, the Jasper Series can help students to become independent thinkers and learners rather than simply able to perform basic calculations and retrieve basic knowledge facts. It was noted that students who were in Jasper classes scored better on word and planning problems than the children who were in comparison classrooms. In addition, it was stated that the Jasper Series had a very positive effect on children’s attitudes toward mathematics and problem solving (The Cognition and Technology Group at Vanderbilt).
According to NCTM (2003), in grades K-2, students’ progress in mathematics should be assessed in order to understand what they can do or cannot do. To get more accurate information about what children can or cannot do, the NCTM provides five mathematics assessment standards that are considered as “learning targets” or “benchmarks for learning” (p. 9), and that can be used for monitoring students’ progress, making instructional decisions, and evaluating students’ achievement at the end of each unit of instruction:

- the learning standard
- the equity standard
- the openness standard
- the inferences standard
- the coherence standard

First, effective assessment, which is considered a part of instruction, should provide opportunities for students to illustrate what they know and can do; thus, assessment’s contribution to students’ learning of mathematics and its relationship to instruction should be determined carefully to enhance learning. Second, assessment should promote equity. Each child has unique qualities and experiences, thus equitable assessments would provide opportunities for children to demonstrate their unique power. Third, assessment should be an open process. According to the NCTM, before being assessed, students and the public (including parents, policymakers, business and industry leaders, members of the mathematics community, and interested citizens) should be informed about the nature of the assessment so that everyone who is impacted by the assessment of students’ learning has the opportunity to develop a collective understanding of the performance criteria for mathematics. Fourth, assessment should promote valid inferences about students’ mathematics learning. Finally, assessment should be a coherent process, which means that assessment should match the purposes for which it is being done and should be aligned both with the curriculum and with instruction. The five assessment standards provided by NCTM serve as important criteria that can be used to inform and improve the quality of software’s assessment components.
Feedback

Extrinsic and intrinsic feedback is essential in educational software because it represents the effects of the learner’s action on the system or environment (Squires & Preece, 1999). Feedback might include answer correctness, learning guidance, timeliness, motivational messages, and learning focus (Hoska, 1993; Margolis & McCabe, 2003; Sales, 1993; Skinner, 1958). According to the National Council of Teachers of English (NCTE, 2001), feedback should not only provide the right answer but also enable learners to enhance their skills and understanding. Deubel (2002) suggested that feedback as well as the question to which the feedback applies should be on the same screen to reduce the memory load on students. Also, Deubel stated that learners should have only two attempts to answer the question. If both answers are incorrect, the program should provide the correct response and indicate why the answers were wrong.

There is disagreement regarding the preference for immediate feedback versus delayed feedback. Scholars who support the effectiveness of immediate feedback point out that if the program provides immediate feedback following students’ response, students are better able to adjust and improve their knowledge as they are still focused on the subject (DeLaurentiis, 1993; Johnson & Johnson, 1993; Khalifa, Bloor, Middleton, & Jones, 2000; Kulik & Kulik, 1988; Skinner, 1954, 1958; Swenson & Anderson, 1982). Skinner (1958) drew attention to the importance of immediate feedback stating:

…of course, the machine, like a private tutor, reinforces the student for every correct response, using this immediate feedback not only to shape his behavior most efficiently but to maintain it in strength in a manner which the layman would describe as holding the student’s interest. (p. 971)

Skinner (1958) also argued that allowing a few seconds between a student’s response and feedback from the program reduces most of the positive effects of feedback. In contrast, immediate feedback may interrupt the flow of interaction and affect learners’ motivation negatively (Alty, Bergan, Craufurd, & Dolphin, 1993). Other researchers argued that delaying the feedback for a day or more appeared to increase what a student remembered on a retention test, and students who received delayed feedback spent more time repeating the item than did other students who received immediate feedback (Brackbill, Bravos, & Starr, 1962; Kulhavy, 1977; Kulhavy & Anderson, 1972).
With regard to the importance of feedback, software should give positive reinforcement for the submission of correct answers. For example, statements like “good job,” “you did it,” or “you can do this one more time” are encouraging to students. On the other hand, when children submit an incorrect response, the program should allow them to try to solve the problem at least one more time. If the user submits an incorrect answer again, then the software should be designed to help children understand why their answer was incorrect. Briefly, children should have two or three attempts to answer a question, and if all answers are incorrect, then the software should provide feedback that explains why the answers were wrong further supporting the users’ skills and understanding.

Additionally, feedback given as a part of formative assessment helps learners become aware of any gaps that exist between their desired goal and their current knowledge, understanding, or skill and guides them through actions necessary to obtain the goal (Ramaprasad, 1983; Sadler, 1989). For instance, software may provide a progress report at the end of each session that shows children the number of correct and incorrect answers and provides suggestions for children to draw their attention to challenging tasks or topics.

CHILDREN’S PERSPECTIVE

Recently, scholars have begun to create software with design input from the children for whom they are creating the software (Beyer & Holtzblatt, 1998; Bruckman & Bandlow, 2002; Druin, 1996, 1999; Druin, Stewart, Proft, Bederson, & Hollan, 1997; Large, Beheshti, & Rahman, 2002; Stewart, Raybourn, Bederson, & Druin, 1997). According to the scholars cited, children should be regarded as learners, critics, inventors, and technology design partners and consulted during the process of educational software design.

It is essential for designers to collaborate with children because, as Druin (1996) pointed out, children have their own likes, dislikes, and needs, which are completely different from those of adults. Druin also suggested that developers listen to, and observe, and work with children as they develop software to better address children’s unique needs. With regard to the view that a partnership between users and designers is necessary, the following
section will focus on children’s preferences for design criteria of educational software.

Content of the Software

As noted earlier, software should be both educationally beneficial and entertaining (Costabile et al., 2003; NAEYC, 1996). Tammen and Brock (1997) carried out a study with children between 5 and 14 to learn about children’s expectations. Some of the children in the study complained that the software they used was boring, repetitive, and inappropriate for the age level suggested by the designer of the software.

Visual Environment

Children want to access pictures, videos, and sounds as they use software (Druin, 1997; Large et al., 2002; Tammen & Brock, 1997). For instance, the child participants in Tammen and Brock’s study wanted to see interesting and age-appropriate graphics rather than primary school images (Tammen & Brock, 1997). Ju and Cifuentes (2002) conducted a study to determine the influence of 3D-computer animations on students’ understanding of color theory. Participants in the study were 110 first grade students from two elementary schools. The child participants watched the presentation twice. The first time, they did not discuss the presentation. However, after the second one, children discussed the color names with each other. Also, before and after the computer animation children took a test and its results were compared by the nonparametric test, Wilcoxon signed rank. Based on the results, Ju and Cifuentes concluded that 3D-computer animations have a significant effect on children’s learning since all participants performed the same on both pretest and posttest and they improved their understanding of the results of color mixing. Ju and Cifuentes agreed that these animations are an effective teaching and learning tool since children can gain skills described in state standards by using software that includes 3D-computer animations.

On the other hand, two studies have provided evidence that children are distractible (Halgren, Fernandes, & Thomas, 1995; Hanna, Risden, &
Alexander, 1997). Halgren et al. drew attention to icons on the screen and contended that children would click these icons to see what would happen and if one of them made a sound or became animated, the children would keep clicking it. Likewise, Hanna et al. (1995) used a funny noise that children could hear when an error message popped up and the researchers found that children repeatedly clicked the wrong answers in order to hear the noise.

Feedback and Assessment

Studies found that children also consider feedback an essential part of learning. For instance, researchers found that students appreciated the instant nature and speed of the feedback provided in the course of a computer-based test (Beevers et al., 2002). Likewise, Large, Beheshti and Rahman (2002) found that while most of the students in the study were doing research about a specific topic, children expressed their preference for immediate answers in the form of matching hints instead of being asked new questions. Thus, feedback should be considered as an important factor that influences students’ success (Margolis & McCabe, 2003; Squires & Preece, 1999).

CONCLUSION

Today, computer mediated learning has became more prevalent in day care centers, preschool programs, elementary schools, community centers, and after-school programs. Computers and educational software have the potential for enabling young children to develop, to broaden, and to deepen their mathematical knowledge. This increased knowledge and competency in mathematics can significantly enhance their future educational and career opportunities. To increase the likelihood that computers and computer programs will have a positive impact on student learning, designers should collaborate with both researchers and children to ensure that children are provided with technology-enriched learning experiences that are developmentally appropriate. At the beginning of the design process, designers should identify the purpose of the software, and then decide which topic he/she needs to cover to address children’s unique needs. To this end, designers
may benefit the NAEYC recommendations and the NCTM principals and standards. To support children’s understanding of mathematical concepts covered in the software, the software should include assessments and practices that enable children to monitor their progress. Designers should also consider the visual environment in the software. Since the recent studies showed that colorful animations, pictures, songs, and symbols can keep children more focused on a subject, it is crucial for designers to include developmentally appropriate visual and auditory tasks for children. However, software should be free of violent images or content in order to promote children’s intellectual, physical, social, spiritual, emotional, and moral development. Further investigations examining the software design processes in more detail should help to understand what should be initiated into the software to facilitate children’s mathematical growth.

References


Deubel, P. (2002, February). Selecting curriculum-based software: Valuable educational software can help students rise to the challenge of
standardized testing and assessment. Learning and Leading with Technology, 29(5), 10-17.


