Designing Tools for Developing Minds: The Role of Child Development in Educational Technology

Amanda Strawhacker
Dylan Portelance
Melissa Lee
Marina Bers
Developmental Technologies
Research Group
Tufts University
Medford, MA 02155, USA
Amanda.Strawhacker@tufts.edu
Dylan.Portelance@tufts.edu
Melissa.Lee@tufts.edu
Marina.Bers@tufts.edu

Abstract
In today's world, in which technology is playing an increasingly growing role in the lives of children, computer literacy and technological fluency are becoming a necessary national standard. However, developing character traits that serve children to use new tools in a safe way to communicate and connect with others, and providing opportunities for children to envision a better world through the use of technology is just as important. The DevTech Research Group is concerned with developing technologies, interventions, and evaluations within the framework of children's positive technological development. This paper presents key design principles for researchers investigating educational best practices investigating educational best practices in technology-rich learning environments.

Author Keywords
early childhood; computer programming; educational technology; child development, positive technological development; computational thinking.

ACM Classification Keywords
K.5.m [**Computers and Education**] Miscellaneous – *Computer literacy*

Introduction
Coding, computational thinking, and digital literacy have received much attention the K-12 education community, with policy makers and researchers generally agreeing these skills should be taught in schools, alongside other academic disciplines and

Figure 1: Two kindergarten girls share their coded creations using the ScratchJr programming environment. ScratchJr was created in part by the DevTech Research Group.

domains of learning. Although there is a diversity of opinions about coding in the classroom, many non-profit organizations http://www.code.org, national education associations 8, and government offices 4, 13 have publicly declared support for technology education for children. In their statement on technology for children from birth through age 8, the National Association for the Education of Young Children (NAEYC) states that "when used appropriately, technology and media can enhance children's cognitive and social abilities." But despite NAEYC's well-researched cautions about what to avoid when using technology with children, there is still little advice about what it means to use technology appropriately with young children. In this paper, we offer principles for designing developmentally appropriate technological experiences for young learners.

Developmentally Appropriate Technologies

Seymour Papert inspired a new generation of educators with his LOGO programming language and his vision for a computer-literate learning program for children. His philosophy, constructionism, set the tone for researchers who would use technology as an expressive tool, a "tool to think with" 9. Our research at the Developmental Technologies (DevTech) Research Group is rooted in this body of research, as well as that of positive youth development, as conceptualized by applied developmental sciences 6. The fusion of these theoretical perspectives has resulted in the development of the Positive Technological Development (PTD) framework by Bers 1 for understanding the ways that technology can positively influence children's development. Now, DevTech applies these grounding theoretical frameworks to a) new research on educational technology for young children and b) the development of new technological tools, curricula, and teaching resources for use in early childhood http://ase.tufts.edu/DevTech/. In this paper, we describe several principles that guide our research, technology development, and interventions with very young children.

Principles to Guide Educational Technology Design for Young Children

These statements illustrate the pedagogical approach used at the DevTech Research Group to design developmentally appropriate programming and engineering experiences for young children.

Technology should be intentionally designed to leverage what we already know about children's development. Much is known about how children develop and grow over time, how to design effective environments for learning, and how children engage with non-digital educational tools 5, 9, 12, 15. Current research in child development, as well as teaching perspectives such as Developmentally Appropriate Practices (DAP) for teaching and learning 7 should be considered in the initial design of any educational technology. For example, if we already know that fine motor movements and icon recognition skills are still developing in most 6-year-old children, it makes sense to design a tool with larger buttons and simpler images, rather than arbitrarily applying ergonomic principles and confusing icons (i.e. floppy disk for a save icon) that are better suited to adults. When creating technological spaces that are inviting, exciting, and appropriately challenging for children, we should consider children's unique limits, needs, and capabilities.

Children should direct their own playful learning experiences with technology – not the other way *around.* We often use the juxtaposition of two physical learning spaces, playpen and a playground, to explain the nuances of different digital educational environments 1. Although many technologies use a "drill-and-kill" style to help children memorize facts from a single domain chosen by the tool's creator 9, there are a few digital spaces that are open-ended and cross-domain, allowing for creativity, exploration, and even mistakes. Coding languages provide a uniquely low-stakes, high-reward learning environment for children to create anything they can imagine, from an animated story to a robotic dance. Just as with physical playground spaces, digital playground experiences should allow for the development of a broad range of children's learning domains, such as social collaboration, physical movement, and identity formation. Tools and curricula should encourage children to explore the creative, expressive aspects of programming and engineering.

Technologies should be used to educate children about technology itself. The world in which children are being raised today is drastically different than it was even 20 years ago. Children are immersed in environments with wearable technology, smart phones, sensor-activated appliances, and computers, and yet almost no early childhood classrooms spend time exploring these apparently magical devices. Programming allows children to think about concepts of sequencing, order, and logic in approachable ways, and to apply these foundational concepts to any technology they encounter in their world. The role of any learning tool is to help children bridge the gap between what children can understand without guidance, and what they can achieve with it 15. In that sense, coding languages for children are specialized tools that allow children to break down abstract concepts and see their constituent (manageable, concrete) parts 14. Well-designed technology should help children to identify programmatic experiences in their every day lives, through the lens of the playful coding language they learned on.

Foundational concepts of programming and engineering should reinforce key concepts in other learning domains. For a child, the main benefit of learning to code is not to master a coding language, or even to prepare for a future career in programming. In its simplest form, a coding language is an instrument that takes an abstract concept, like a system of procedures, and makes it concrete. Because coding offers a series of observable cause-and-effect actions, it can be a platform for playing with abstract ideas of sequencing, order, rotation, etc., even for the most literal thinkers: Kindergarten through second grade children. More importantly, the abstract ideas behind coding are more powerful than the coding language itself 9, and can extend to other domains. Research has shown that children who engage in classroom-based programming interventions, even brief ones, display significant improvement in foundational sequencing skills rooted in early literacy and life skills, including picture-based storytelling tasks and kinesthetic executive functioning tasks 7. Brennan and Resnick 3 discuss the ways that coding affect children's computational thinking skills, from introducing cross-domain concepts sequencing and loops, to developing computational practices like debugging and iterative design, extending even to the development of new thinking perspectives, such as that of an inquisitive re- designer, or an expressive creator.

A well-designed technology is only as good as the environment in which is it used. The teaching opportunities listed above can be seized or missed, depending on the implementation of the technology. No tool is inherently educational. Its effectiveness depends upon the context in which it is introduced. The curriculum, the classroom culture, the individual student, and the myriad other factors that go into a child's development all play a role in how the child understands a technological experience. For this reason, it is imperative that we continue to research learning outcomes, develop curricula, and collaborate with education policy makers in order to successfully integrate programming and engineering into children's lives.

Conclusion

Our design research is strongly inspired by the Positive Technological Development theoretical framework. We focus on both technologies and educational materials and pedagogies. It will be exciting to see how future work examines these principles as they apply to newer and more diverse programming tools, and even more exciting to follow the new principles and perspectives as they develop. Through the combination of theory and innovation, the DevTech Research Group hopes to develop new experiences for promoting learning, not for technology's sake, but for the sake of children learning to explore the world through a different lens.

Acknowledgements

We thank all the researchers, schools, families, and funding bodies that have aided in the DevTech Research Group's continued contributions to the field of educational technology.

References

1. Marina U. Bers. 2012. *Designing digital experiences for positive youth development: From playpen to playground.* Cary, NC: Oxford.
2. Marina U. Bers. 2010. The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research and Practice* 12, 2.
3. Karen Brennan and Mitchel Resnick. New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (AERA '12). http://web.media.mit.edu/~kbrennan/files/Brennan_Resnick_AERA2012_CT.pdf
4. Department for Education. 2013. The National Curriculum in England: Framework document. London: The Stationery Office.
5. Gabriel Guyton. 2011. *Using toys to support infant-toddler learning and development*. NAEYC: Young Children, p.51.
6. J. V. Lerner, E. Phelps, Y. E. Forman, and E. P. Bowers. 2009. Positive Youth Development. *Handbook of Adolescent Psychology*. 1:II:15.
7. Elizabeth Kazakoff, Amanda Sullivan, and Marina U. Bers. 2013. The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal* 41,4: 245-255. doi:10.1007/s10643-012-0554-5.
8. National Association for Education of Young Children. 2009. *Developmentally Appropriate Practice in Early Childhood Programs Serving Children from Birth through Age 8.*
9. National Association for Education of Young Children and Fred Rogers Center for Early Learning

and Children’s Media. 2012. *Technology and Interactive Media as Tools in Early Childhood Programs Serving Children from Birth through Age 8.* Washington, DC.

10. Seymour Papert. 1980. *Mindstorms: Children, Computers, and Powerful Ideas.* New York, Basic Books.
11. Jean Piage and Bärbel Inhelder. 1969. *The Psychology of the Child*. New York: Basic Book.
12. Mitchel Resnick. 2006. Computer as paintbrush: Technology, play, and the creative society. In Singer, D., Golikoff, R., and Hirsh-Pasek, K. (Eds.), *Play = learning: How play motivates and enhances children’s cognitive and social-emotional growth.* Oxford University Press.
13. Ellen Y. Stevens. 1912. Montessori and Froebel: A comparison. *The Elementary School Teacher* 12, 6: 253-258.
14. US Department of Education, Office of Educational Technology. 2010. Transforming American education: Learning powered by technology. Draft National Educational Technology Play 2010. Washington, DC.
15. Bret Victor. Up and Down the Ladder of Abstraction. Retrived March 15, 2015. http.//worrydream.com/LadderOfAbstraction/
16. Lev Vygotsky, R. W. Rieber, and Aaron S. Carton. 1987. *The Collected Works of L. S. Vygotsky*. New York, Plenum Press.