International Journal of Child-Computer Interaction ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

International Journal of Child-Computer Interaction

journal homepage: www.elsevier.com/locate/ijcci

The Seymour test: Powerful ideas in early childhood education

Marina Umaschi Bers

Tufts University, United States

ARTICLE INFO

Article history:
Received 18 January 2017
Accepted 26 June 2017
Available online xxxx

"When one enters a new domain of knowledge, one initially encounters a crowd of new ideas. Good learners are able to pick out those which are powerful." Seymour Papert, 1980.

I will never forget my first meeting with Seymour Papert. It was in the early 90s' and I had recently arrived to Boston from Argentina. I was nervous. I was about to meet the "great" Seymour Papert. As a child, back in the late seventies, my parents had sent me to learn LOGO through a pilot project run by IBM in my hometown of Buenos Aires. Decades later, I could not believe that I was at the MIT Media Lab and that my mentor was going to be the father of LOGO.

I had prepared lots of questions and several sketches of ideas to discuss with Seymour. I am a visual person, so I had a notebook full of diagrams to share with him. However, as soon as I got to his office, he told me that we needed to go to the supermarket to get some groceries he had forgotten. So there we went. While choosing tomatoes and cheeses, we were able to more or less engage in a meaningful conversation.

It was not at all the way I had planned it. I could hardly understand Seymour's South African accent, and he probably could not understand my thick Argentinean accent either. I could not show him my diagrams, since we were constantly walking the aisles of the supermarket. However, the background noise made it less awkward to ask each other to repeat our sentences several times. It forced us to take the time and pause often. Seymour wanted to understand me. Do not get me wrong. I do not think he was interested in getting to know me. He wanted to understand my ideas.

Seymour was a man of ideas. He fell in love with computer programming because of its potential to bring about new ideas. Both at the personal and the societal level. Ideas can change the world. Having grown up with the Apartheid, Seymour wanted to change the world. However, ideas are not easily grasped. They are abstract. They are not tangible. They are difficult to share. They need to be discovered through hard work. Objects facilitate this process. Seymour understood the importance of "objects to think with". These objects can help us make ideas concrete, tangible and sharable. They also empower us to have new ideas or to see old ideas in new ways. Objects can exist on the digital screen or on the physical world. Seymour knew about this early on. The LOGO turtle was both a virtual cursor and a floor-based robot.

In his seminal work *"Mindstorms: Children, Computers and Powerful Ideas"* [1], Seymour shares his big ideas about how children can become better learners and thinkers through coding. However, although the book became popular and much cited, Seymour expressed a sense of frustration. On the title, readers focused too much on the word "computers", and not enough on "powerful ideas".

Powerful ideas are one of the most complex concepts to understand within Papert's Constructionist approach to education and probably one of the most intellectually rewarding. He coined the term "powerful ideas" to refer to a central concept and skill within a domain that is at once personally useful, epistemologically interconnected with other disciplines, and has roots in intuitive knowledge that a child has internalized over a long period of time. According to Papert, powerful ideas afford new ways of thinking, new ways of putting knowledge to use, and new ways of making personal and epistemological connections with other domains of knowledge [2].

Over the years, a growing community of researchers and educators have used the term powerful ideas to refer to a set of intellectual tools worth learning, as decided by a community of experts in each of the fields of study [3]. However, different people apply the term in diverse ways and amongst the powerful ideas community there are divergent opinions about the benefits and dangers of presenting a unified definition [4].

The importance paid by Seymour and colleagues to the world of ideas and thinking derives from the Piagetian legacy concerned with epistemology: understanding how we know what we know, how we construct knowledge and how we think. Back in the 60's Seymour envisioned the computer as a powerful carrier of new and old powerful ideas and as an agent of educational change. Now a

E-mail address: marina.bers@tufts.edu.

http://dx.doi.org/10.1016/j.ijcci.2017.06.004

day, with the growing number of available programming environments and robotic kits for children, we can say that Seymour was into something. However, although he became known as the father of LOGO, the emphasis was never on the tools. He admonished us against a technocentric perspective that puts technology at the center-stage [5]. The focus was on ideas.

Seymour talked about three different roles that computers could play in relationship with powerful ideas. They could be neutral, liberators or incubators of powerful ideas [4]. The *neutral role* refers to the fact that some powerful ideas are independent of the existence of the computer. They were installed in the culture long before the computer entered the scene and continue to exist without having experienced major changes. The *liberator role* refers to powerful ideas that existed before the computer but the computer liberated them but making them more powerful and accessible to a wide range of people. Papert uses modeling as an example. The third role that computers can play is as *incubators* of powerful ideas. This refers to the small subset of ideas that are really born from the existence of the computer. Ideas that we could not get to know without the computer.

Seymour believed one of the best uses of the computer was for children to become creators and producers of their own projects through programming. Coding enables children to think in systematic and sequential ways while encountering powerful ideas from computer science and other domains of knowledge. Furthermore, coding can help children think about their own thinking. Given the current focus on promoting executive function and meta-cognition in schools, Seymour was a visionary. However his ideas are still not fully understood by the educational system. Most of the time, when coding is introduced in the formal schooling, it is with the goal to feed the work pipeline or to meet the requirements of increasing demands to reinforce STEM. Coding as epistemology, coding as a way of knowing, is seldom explored.

1. Kittens and wooden blocks at the coding playground

From Seymour I learned the importance of powerful ideas. Every time I engage in a new project, either writing a curricular unit, designing a programming language or a robotic kit or creating content for an animated TV show, I first ask the question: What are the powerful ideas that young children will encounter when experiencing this? I always spend time dwelling in the domain to make sure I can truly identify the most powerful ideas. I learn from others and I read as much as I can. I study how the chosen powerful ideas could become developmentally appropriate for the age of the children I am designing for. Finally, I take time playing with the specific design constraints of the medium to understand how its features can help children best discover those powerful ideas. For example, a tangible programming language with which children can make a robot dance, such as KIBO, affords different explorations than a short TV episode that puts children in the viewer seat.

During the last ten years I have focused my work on young children: 4 to 7 years old. Two [6,7] – document the significance of early experiences for later school achievement. Research shows that both from an economic and a developmental standpoint, educational interventions that begin in early childhood are associated with lower costs and more durable effects than interventions that begin later on (e.g., [8,9]). Research and policy changes over the recent years have brought a newfound focus on STEM education for young children [10,11]. In particular, the release of new learning standards and best practices for integrating technology into early childhood education ([12–16]). As part of this effort, there are new initiatives for introducing computational thinking and coding in early childhood education [17]. I have contributed to this national and international effort in different ways.

Inspired by Seymour's vision to create "objects to think with", I engaged in the design of two programming environments that are developmentally appropriate for young children: ScratchJr, a free app that runs on tablets, and KIBO robotics, a robot kit that can be programmed by putting together sequences of wooden blocks, — without requiring screen time from PCs, tablets, or smartphones. Children can build their own robot, program it to do what they want, and decorate it with art supplies.

ScratchJr was designed and developed by my DevTech research group at Tufts University in collaboration with Mitch Resnick's LifeLong Kindergarten group at the MIT Media Lab and Paula Bonta and Brian Silverman from the PICO company in Canada [18]. To date over six million young children all over the world are using ScratchJr to create their own projects and it has been translated to several languages (see Fig. 1). Furthermore, in December 2015 an extension was launched in collaboration with PBS KIDS. Children are now able to also create their own projects using their favorite cartoon characters from popular TV shows.

Inspired by the popular Scratch [19], the powerful ideas behind ScratchJr are simple: an introductory programming language, designed to be developmentally appropriate, that enables young children (ages 5 to 7) to create their own interactive stories and games. Children snap together graphical programming blocks to make characters perform various actions (e.g., move, jump, dance, and sing). Using ScratchJr does not require the ability to read or write. It was designed to reinforce and train abilities like sequencing, understanding of causality, and problem solving while engaging in open-ended creative projects [17].

With ScratchJr, coding happens on the screen. With KIBO robotics, it happens with blocks, without screens of any kind. The concept, prototypes and research for KIBO were born in my Developmental Technologies Research Group at Tufts University back in 2011 through generous funding from the National Science Foundation (NSF). KIBO became commercially available worldwide in 2014 through KinderLab Robotics (see www.Kinderlabrobotics.com) The powerful ideas behind KIBO are similar to ScratchJr, however the key concept is to provide a coding environment in which children can engage in similar experiences as they do in a playground [20]. They can use their bodies, as well as their minds.

The KIBO programming language is composed of wooden blocks with pegs and holes that can be inserted into each other forming a tangible sequence of commands. Each block represents an instruction for KIBO: forward, shake, wait for clap, light on, beep, etc. Designed with a playground approach, KIBO supports children in making almost anything: a character from a story, a carousel, a dancer, a dog sled. The possibilities are endless, as wide as children's own imaginations [17,21,22]. The child puts together a sequence of instructions (a program) using the wooden KIBO blocks. Then, they scan the blocks with the KIBO body to tell the robot what to do. Finally, they press a button and the robot comes "alive". KIBO engages children in becoming programmers, engineers, problem solvers, designers, artists, dancers, choreographers and writers (see Fig. 2).

Both KIBO and ScratchJr were inspired by the rich tradition of constructionist learning environments initiated by LOGO. I describe myself as a designer of playground experiences for coding [17]. The fundamental powerful idea I work with is the understanding of the developmental characteristics of young children and the context in which they are likely to engage with programming.

I coined the metaphor "playground vs. playpen" [20] to discuss the role that new technologies can have in young children's lives. Playgrounds are open-ended. Playpens are limited. Playgrounds invite fantasy play, imagination and creativity. The "playground vs. playpen" metaphor provides a way to understand the kind of developmentally appropriate experiences that programming languages can promote: problem solving, imagination, cognitive challenges, social interactions, motor skills development, emotional

Please cite this article in press as: M.U. Bers, The Seymour test: Powerful ideas in early childhood education, International Journal of Child-Computer Interaction (2017), http://dx.doi.org/10.1016/j.ijcci.2017.06.004.

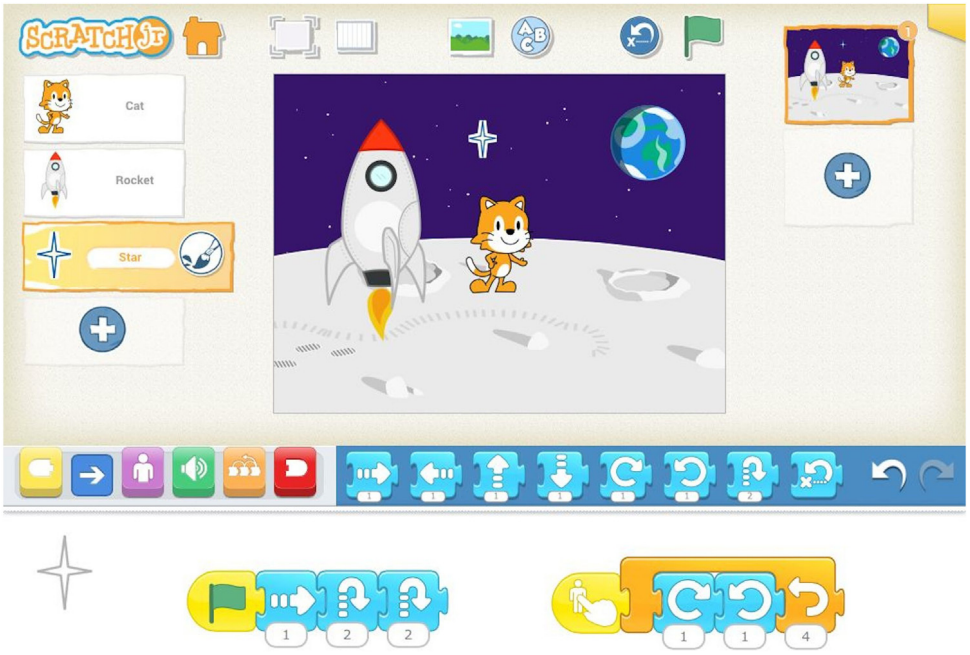

Fig. 1. A screenshot of the ScratchJr app.

exploration, and making different choices. In contrast to the open-ended playground, playpens convey lack of freedom to experiment, lack of autonomy for exploration, lack of creative opportunities, and lack of taking risks. Although playpens are safer, playgrounds provide infinite possibilities for growth and learning.

By designing environments such as ScratchJr and KIBO, my goal was to promote the idea that coding can become a playground. An environment to be creative, to express ourselves, to explore alone and with others, to learn new skills and problem solve. An environment to explore powerful ideas. All of this, while having fun. Within the playground approach, I am concerned with promoting positive youth development, and not only with improving problem solving and mastering the art of coding.

In the coding playground, young children create their own projects to communicate ideas and express who they are. They need developmentally appropriate tools, such as KIBO and ScratchJr. They engage in problem solving and storytelling, they develop sequencing skills and algorithmic thinking. They journey through the design process from an early idea to a final product that can be shared with others. They learn how to manage frustration and how to persevere towards finding a solution, rather than giving up when things get challenging. They develop strategies for debugging or fixing their projects. They learn to collaborate with others and they grow proud of their hard work. In the coding playground, children have fun while learning new things. They can be themselves and playfully explore new concepts and ideas, as well as develop new skills. They can fail and start all over again. In this coding playground children encounter powerful ideas from computer science that are useful not only for future programmers and engineers, but for everyone. Coding as a literacy for the XXIs century.

2. The Seymour test

Designing KIBO and ScratchJr was a lot of fun. However, throughout the process I kept asking myself: will these tools pass what I am calling " The Seymour test"? Will he understand the powerful ideas I was trying to convey? What guided my efforts in the design of these programming environments was the goal

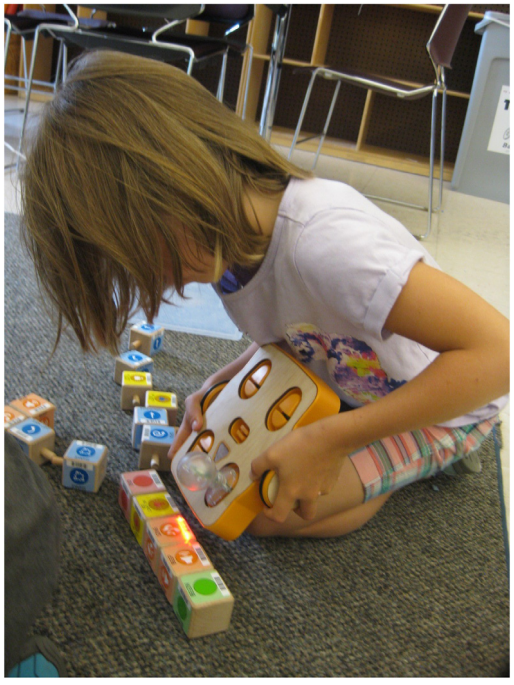

Fig. 2. Programming KIBO.

to help children code in a playful way so they could engage in computational thinking. So they could encounter powerful ideas. Amongst them, one became the most salient and developmentally appropriate for this age range: sequencing.

Sequencing is indeed a powerful idea that I believe can pass the "Seymour test". It is *personally useful* to understand that there is an orderly set of steps to follow when trying to achieve a certain outcome (i.e cooking). It is *epistemologically interconnected* with other disciplines, sequencing is the basis for algorithm in computer science, counting in math, reading and writing, etc. It has *roots* in

Please cite this article in press as: M.U. Bers, The Seymour test: Powerful ideas in early childhood education, International Journal of Child-Computer Interaction (2017), http://dx.doi.org/10.1016/j.ijcci.2017.06.004.

intuitive knowledge that a child has internalized over a long period of time. For example, four year olds know the sequence of time that dictates day and night, the weekly calendar and the four seasons. In ScratchJr sequencing is explored by connecting blocks on the screen, in KIBO by putting together wooden blocks.

Back in the late 90's, when I was a doctoral student working with Seymour at the MIT Media Lab, the joke was that Seymour did not come in the LOGO box. What we meant was that, although we were bringing LOGO and its full expressive potential to schools, many teachers tended to use LOGO in traditional instructionist ways. Creativity and personal expression were left out, as well as the concept of powerful ideas. Our internal joke hid our wish to have Seymour come inside the LOGO box, so he would come out with his charming personality and help us convince teachers to let their students freely explore and create a project of their choice and encounter new ideas.

We spent hours sharing his theory, his philosophy, and pedagogical approach with as many teachers as we could find. Seymour did not come in the LOGO box, but Constructionism, the framework he developed, came in the book *"Mindstorms: Children, Computers and Powerful Ideas"* [1]. Years later, when I became a faculty at Tufts University with an appointment in the Child Development department, I took upon myself the mission of bringing Constructionism to early childhood education. Most of the powerful ideas about learning and technology that Seymour had struggle to convey and were encountered with resistance in formal schooling in the upper grades, were welcomed and already present in early childhood education: learning by doing; learning by playing; learning by making;; learning with objects; learning with a community; learning by reflecting [3]. From there, it is easy to see how young children can also learn by coding. Tools that are developmentally appropriate, such as ScratchJr and KIBO, make that possible.

Constructionism is my intellectual home. It is from there that I developed my own theoretical framework which I call Positive Technological Development (PTD) [3,20]. PTD examines the developmental tasks of children growing up in the digital era and provides guidelines to design and evaluate technologically-rich environments that support positive behaviors: Communication, Collaboration, Community-building, Content creation, Creativity and Choices of conduct.

In my work I have translated Seymour's concept of powerful ideas to the world of early childhood education. My goal is to make sure that as new technologies start to enter preschool and kindergarten, powerful ideas are not left behind. Neither are misunderstood.

In particular I found useful to make the distinction with the concept of wonderful ideas proposed by Eleonor Duckworth [23]. Wonderful ideas are the essence of intellectual development and are expressed as personal insights or revelations. Following a Piagetian tradition, in Duckworth's vision, wonderful ideas are deeply connected with the developmental stage of the individual and the stepping into a new stage. Wonderful ideas are results of an individual's previous knowledge combined with intellectual alertness to ask new questions and play with materials in new ways.

Although Papert's powerful ideas and Duckworth's wonderful ideas have many things in common, they stress slightly different dimensions of learning. Whereas Duckworth's wonderful ideas refer to the developmental process of an individual, Papert's powerful ideas take a cultural and epistemological perspective. Certain ideas are very powerful, and children should be given the chance to explore them. While both wonderful and powerful ideas are grounded on personal excitement and confidence to try them out, wonderful ideas do not have the meme-like quality to become powerful in themselves and spread into a culture and historical time. Once upon a time, all the powerful ideas were wonderful ideas that started with an "A-ha moment". But all wonderful ideas will not become powerful ideas. Powerful ideas are wonderful ideas that stand the test of time and that are successful in the marketplace of ideas.

Back in 2005, I wrote to Seymour to ask if he would be wiling to write the preface for my book "Blocks to Robots: Learning with Technology in the Early Childhood Classroom" [3]. In my email I explained to him how " this book will be a wonderful way to get the idea of Constructionism out into the world of early childhood educators". I was scared of his response. Seymour was not polite when he was not happy about something. I also knew that in the past he had refused to give a concrete definition of Constructionism. In 1991, he wrote, *"It would be particularly oxymoronic to convey the idea of constructionism through a definition since, after all, constructionism boils down to demanding that everything be understood by being constructed".* [24]. I explained to him, that I would not provide a definition but I would highlight how the powerful ideas of Constructionism were relevant and well-aligned with the concept of developmentally appropriate practice in early childhood education. Furthermore, I was well-aware of Seymour's sometimes aversion to write specifically for a teacher's audience. And my book was for early childhood educators. He believed in learning, much more than in teaching.

Two days after I sent him my request to write the preface for the book, I got his answer.

From: Seymour Papert <papert@media.mit.edu>
Subject: Re: question
Date: April 5, 2006 at 2:18:31 AM EDT
To: "Bers, Marina" <Marina.Bers@tufts.edu>
Cc: Seymour Papert <papert@media.mit.edu>

I would be delighted to do it.

I was trilled. I had passed Seymour's test. His answer showed trust that I would do a good job conveying the powerful ideas of Constructionism to this new audience. On December 5, 2007, Seymour was severely injured in a traffic accident in Hanoi, Vietnam. There was no preface. There was no more the Seymour we all knew. However, his powerful ideas are still empowering us through our learning journeys.

References

[1] S. Papert, Mindstorms: Children, Computers and Powerful Ideas, Basic Books, New York, 1980.
[2] S. Papert, What's the big idea? Toward a pedagogy of idea power, IBM Syst. J. 39 (3/4) (2000) 720.
[3] M. Bers, Blocks To Robots: Learning with Technology in the Early Childhood Classroom, Teacher's College, New York, 2008.
[4] S. Papert, M. Resnick, Rescuing the powerful ideas, in: NSF Symposium, MIT, 1996.
[5] S. Papert, Computer criticism vs. technocentric thinking, Educ. Res. 16 (I) (1987) January/February 1987.
[6] National Research Council, Eager to learn: educating our preschoolers, in: Infant and Child Development, Vol.11, National Academy Press, Washington, DC, ISBN: 0-309-06836-3, 2001, pp. 283–284. http://dx.doi.org/10.1002/icd.271.
[7] National Research Council, From Neurons To Neighborhoods: The Science of Early Childhood Development, National Academy Press, Washington, DC, 2002.
[8] F. Cunha, J. Heckman, The technology of skill formation, Amer. Econ. Rev. 97 (2) (2007) 31–47.
[9] J. Heckman, D.M. Masterov, The productivity argument for investing in young children. Working paper 5, Invest in kids working group, Center for Economic Development. 2004.
[10] Sesame Workshop. Sesame workshop and the PNC Foundation joint White House effort on STEM education. 2009. Retrieved from http://www.sesameworkshop.org/newsandevents/pressreleases/stemeducation_11212009.
[11] White House. Educate to innovate, 2016. Retrieved from https://www.whitehouse.gov/issues/education/k-12/educate-innovate.
[12] US Department of Education. Educate to innovate. 2010. Retrieved from http://www.whitehouse.gov/issues/education/k-12/educate-innovate.

Please cite this article in press as: M.U. Bers, The Seymour test: Powerful ideas in early childhood education, International Journal of Child-Computer Interaction (2017), http://dx.doi.org/10.1016/j.ijcci.2017.06.004.

[13] International Technology and Engineering Education Association, Standards for Technological Literacy, third ed, International Technology Education Association, Reston, VA, 2007.
[14] National Association for the Education of Young Children (NAEYC) & Fred Rogers Center. Technology and interactive media as tools in early childhood programs serving children from birth through age 8, 2012. Retrieved from http://www.naeyc.org/files/naeyc/file/positions/PS_technology_WEB2.pdf.
[15] K12 Computer Science Framework 2016. Retrieved from: http://www.k12cs.org.
[16] B. Barron, G. Cayton-Hodges, L. Bofferding, C. Copple, L. DarlingHammond, M. Levine, Take a giant step: A blueprint for teaching children in a digital age. New York, NY: The Joan Ganz Cooney Center at Sesame Workshop. 2011.
[17] M.U. Bers, Coding as a Playground: Programming and computational thinking in the early childhood classroom, Routledge, 2018.
[18] M.U. Bers, M. Resnick, The Official ScratchJr Book, No Starch Press, San Francisco, CA, 2015.
[19] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, Y. Kafai, Scratch: programming for all, Commun. ACM 52 (11) (2009) 60–67 ….
[20] M.U. Bers, Designing digital experiences for positive youth development: From playpen to playground. Cary, NC: Oxford, 2012.
[21] A. Sullivan, M. Elkin, M.U. Bers, KIBO robot demo: Engaging young children in programming and engineering, in: Proceedings of the 14th International Conference on Interaction Design and Children, IDC'15, ACM, Boston, MA, USA, 2015.
[22] A. Sullivan, M.U. Bers, Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade, Int. J. Technol. Des Educ. (2015) Online First.
[23] E. Duckworth, "The Having of Wonderful Ideas" and Other Essays on Teaching and Learning, Teachers College Press, NY, 1996.
[24] S. Papert, I. Harel, Situating constructionism, Constructionism 36 (1991) 1–11.

Please cite this article in press as: M.U. Bers, The Seymour test: Powerful ideas in early childhood education, International Journal of Child-Computer Interaction (2017), http://dx.doi.org/10.1016/j.ijcci.2017.06.004.