The Design of Technological Tools for Thinking and Learning

General Information

Learning Sciences 426 / CS 496
Winter 2021, Wednesday, 2:00-4:50 PM
Zoom Virtual Classroom

Professor

Uri Wilensky
Tel. (847) 467-3818, Annenberg 337
uri@northwestern.edu

Teaching assistants

Kit Martin
615-406-3584
kitmartin@u.northwestern.edu
Office Hours Link: https://northwestern.zoom.us/j/3133073028
Office hours: Mondays 9:00 am - 11:00 am

Jacob Kelter
Tel. (575) 639-4474, Cook 5448
jacobkelter@u.northwestern.edu
Office hours link: https://northwestern.zoom.us/j/99254617211
Office hours: Tuesdays 3-5pm

Gabby Anton
Tel. (231) 668-2867, Annenberg 307
gabbyanton@gmail.com
Office hours link: https://northwestern.zoom.us/j/99753956933
Office hours: Thursdays 10am-12 pm

First class: January 13th

Last class: March 17th

Course Web Sites:

http://ccl.northwestern.edu/dtttl/2021LS426/Home.html
https://canvas.northwestern.edu/courses/134011

Email Lists

Course Instructors: cd-fac@ccl.northwestern.edu

Course Members (students and faculty): cd@ccl.northwestern.edu

Most often, we'll send email through the course canvas page.

Slack Channel

We have a class Slack workplace so that we can all communicate more easily. We'll post class announcements in slack in addition to the Canvas course. We hope that you use this to share questions and support peers. TAs will also be present in this space to support students asynchronously. While we are in class, we request that you keep slack open for class participation.

You can join this slack channel via this link: **http://bit.ly/LS426**

Course Description

This course is a hands-on practicum in designing and building technology-enabled curricula and learning environments. We will use many rich software toolkits designed to enable novice computer-users to get their "hands dirty" doing iterative software design. In addition to the hands-on component, the course is also designed to introduce you to the Constructionist Learning design perspective. This perspective, first named by Seymour Papert and greatly influenced by the work of Jean Piaget, is very influential in the learning sciences today. The Constructionist approach starts with the assumption that teaching cannot successfully proceed by simply transferring knowledge to students' heads. Skillful teaching starts with the current state of knowledge of the student. In order for students to learn effectively, they need to construct the knowledge structures for themselves. In the spirit of Constructionism, we will engage in our own construction of artifacts in this class and, through this activity, explore and evaluate the design of kits and tools intended to enable learners to construct their own motivating and powerful artifacts. We will do this by constructing both physical and virtual artifacts and by engaging in reflective discussion of both the artifacts themselves and the tools used to construct them. In the final project, students will put all of this together by designing and implementing a constructionist learning environment.

After completing this course, you should be able to:

1. Design and implement educational software at the prototype level.
2. Design technology-enabled activities that take advantage of the computational medium.
3. Exercise good judgment in such design within the target context, content domain and deployment situation.
4. Avoid common educational software design errors.
5. Assess learning technologies as to appropriateness for educational needs.
6. Evaluate and utilize educational claims of software authors and promoters.
7. Understand the Constructionist design perspective and use it to author and assess software tools and learning environments.

This class will emphasize authoring projects using Logo-like languages. Logo is a computer programming language designed explicitly for use by children and is in use in large numbers of schools, from elementary on up.

Note that *no previous programming background is assumed.*

In fact, the computer languages used in this course are designed to be easy to learn and many thousands of children use them. It is my belief that even if you do not intend to be an educational software designer yourself, it is the reality of today – and more so, of tomorrow – that should inform your choice to become educated about the promise of technology in education. I am confident all of you can learn the programming aspect of the course, as have many students in the past, who had had no prior programming experience. However, programming does take time and you will be expected to devote substantial time to it. This might be frustrating to many of you, initially, but after the first few weeks, you will have the skills you need. You are strongly encouraged to get help from your fellow students through the class canvas forum as well as from the TAs. The TAs will hold weekly office hours designed especially for technical and programming support. We will attempt to schedule these office hours flexibly, and per special requests as well.

In addition to projects, there will be weekly readings: typically, one paper or two short papers per week. There is a considerable literature that we will not have time to read this term. I have provided a more extensive bibliography at the end of the syllabus. You may find some of these extra readings to be useful to you in completing the final project.

Software packages we will use

We will use quite a number of learning software packages in this course. The 3 packages we will use the most are all based on the computer language Logo.
They are:

- Lynx Coding – a multi-media version of basic Logo in common use in elementary schools worldwide. It also includes music, graphics, and web tools. https://northwestern-class-2021.lynxcoding.club/
- NetLogo – a multi-agent version of Logo, this language is tuned for constructing models of complex dynamic systems. It is useful for creating models of ecological systems, chemical systems, economic trade, social behavior, etc. https://ccl.northwestern.edu/netlogo/download.shtml
- NetLogoLab – a NetLogo extension that enables NetLogo to communicate with real world devices such as robots and sensors. We will construct devices that have sensors and motors and can interact with objects in the world (e.g., LEGO robots).

Besides these 3 basic packages, in the software review section of the class, we will also explore a number of other packages. Software we might look at includes: Squeak EToys, Vensim, AgentSheets, Alice, MIT App Inventor 2, Scratch, LEGO Mindstorms EV3, Cubelets, Tuneblocks Impromptu, Tinkerplots, GeoGebra, Sage Modeler, Betty's Brain, Kodu, Pocketcode, Agent Cubes, OzoBot, SPRK+, Strawbees, ToonTalk, CODAP, etc.

Summary of Requirements

This course is designed to be somewhere between a class and a working group. I'm hoping that we'll work together to make sense of readings, and, for most of the class projects, you will be working in small groups.

So the requirements for everyone are:

- Keep up with the readings and participate in class, both in class-discussions during Zoom sessions with your video on and in discussions on Canvas. You will be expected to post a comment on each paper in the week's reading by Tuesday at 5. You will also be expected to keep Slack open during class periods for additional class participation.
- Complete and present several (mostly group) programming assignments using Lynx Coding, NetLogo and NetLogoLab.
- Review one educational software package and present your review in class.
- Design and implement your final project.
- Give a presentation during the last week of the course and present a poster at the class poster session (March 17th, 2021).

In addition, due to the group project nature of the class, you are also asked to send email to cd-fac@ccl.northwestern.edu (as soon as you know) if you cannot make a particular class meeting. You are also responsible for communicating with your project-mates and letting them know in advance if there is any problem with your part of the project.

About the Final Project

The final project is to design and implement a constructionist learning environment. There are two* basic alternatives for this project:

1) Standalone Educational Software (scaffolding in software)
Design and implement some constructionist educational software. This option would involve writing a *design specification* for the software that describes what the software is for, who it serves, why it is needed, why it is best done in software, etc. Subsequent to receiving feedback on the design specification you will need to start working on a *functional specification* of the software itself and then embark on implementing it. You are free to use any authoring tools you like to implement the software as long as you make a good argument for their being well matched to the task. Suggested educational software genres are: a simulation game, a microworld, a collaborative role-play or MUD (a collaborative virtual space or Multi-User Dimension).

2) Software-embedded curriculum (scaffolding in curricular materials)
Design and implement an educational activity that has a computationally embedded component. In this option, you are asked to use one of the three main software environments used in this course: Lynx Coding, NetLogoLab or NetLogo. As above, you would begin with a *design specification*. Depending on the design, you may or may not require a functional specification – it could be a *curriculum flow specification* instead. You would then go on to construct the software and/or Lego constructions that form the kernel of the activity, flesh out the curricular materials that accompany the software and write up a paper that describes one person's (could be yourself) path through the activity.

* For some students, the final project could take a different direction, such as designing a (computational) research model of organizational change using NetLogo. If you're interested in this option, come and talk to me.

Important dates for the final project

- The final project design specification is due by February 24th.
- The final project functional specification (or curricular flow specification) is due by March 2nd.
- The final project is due by March 14th.
- Final projects will be presented on March 17th. You are welcome to invite friends and/or relatives to attend.

Grading

Weekly assignments and projects will be graded as either complete or incomplete. If a project is judged incomplete, you will have an opportunity to complete it or redo it the following week. Often times weekly projects will be presented in the class. You will be assessed on your class presentation and class participation.

Final project:

The final project will be 50% of your grade. If you anticipate having trouble getting the final project completed by March 16th, please contact Professor Wilensky to make arrangements for an extension. Even if you are granted an extension, you must make a coherent presentation on March 17th and present a poster for your project during a public project expo. You will also be assessed on your class participation both in class and virtually.

Readings

Readings will be provided via PDF.

You also need to purchase the following book:

- Papert, S. (1980). *Mindstorms.* New York: Basic Books.

(See the courses website for links to book merchants.)

In general, software projects and reading responses are due on the day before class (Tuesday)
Software projects are due at noon on Tuesdays.
Reading responses are due at 5:00 pm on Tuesdays.

COVID-19 Policy

Due to the COVID-19 pandemic, this class will be held remotely via videoconferencing. Students are expected to attend class virtually at the time class is held. If possible, please keep your video connection on, as it helps facilitate better discussion. We will ask you to participate in the class through various approaches including commenting on shared virtual discussion boards, posting questions in a shared chat, and vocally engaging in discussion. We will discuss these various approaches in more detail in class.

This class or portions of this class will be recorded by the instructor for educational purposes and available to the class during the quarter. Your instructor will communicate how you can access the recordings. Portions of the course that contain images, questions or commentary/discussion by students will be edited out of any recordings that are saved beyond the current term. Unauthorized student recording of classroom or other academic activities (including advising sessions or office hours) is prohibited. Unauthorized recording is unethical and may also be a violation of University policy and state law.

Students requesting the use of assistive technology as an accommodation should contact Accessible NU. Unauthorized use of classroom recordings – including distributing or posting them is also prohibited. Under the University's Copyright Policy, faculty own the copyright to instructional materials – including those resources created specifically for the purposes of instruction, such as syllabi, lectures and lecture notes, and presentations. Students cannot copy, reproduce, display, or distribute these materials. Students who engage in unauthorized recording, unauthorized use of a recording, or unauthorized distribution of instructional materials will be referred to the appropriate University office for follow-up.

Weekly Schedule

Class/Date	Assignments to turn in	Assignments to start	Things to have read (listed on the week they are due)
Class 1 – January 13		Group Quilt Project	"Getting Started", from the DTTTL website
January 19 Class 2 – January 20	Readings Response, 5pm Beginning Lynx (Logo) Activity (Part 1) Group Quilt Project	Lynx (Logo) Coding (Part 2) Lynx (Logo) Coding Hypermedia Project	Papert, S. (1980). *Mindstorms*. (Readings response due the day before, Jan 19th – this will be true for all subsequent reading assignments)
January 26 Class 3 – January 27	Readings Response, 5pm Lynx (Logo) Coding (Part 2)		Harel, I., and Papert, S. (1990). *Software Design as a Learning Environment.* Edwards, L. (1995). *Microworlds as Representations.*
February 2 Class 4 - February 3	Readings Response, 5pm Lynx (Logo) Coding Hypermedia Project	NetLogo or NetLogoLab project (part 1 – extending a model)	Papert, S. (1991). *Situating Constructionism.* Selections from Jean Piaget: Piaget, J. (1952). Conservation of Continuous Quantities
February 9 Class 5 - February 10	Readings Response, 5pm NetLogo/NetLogoLab project (part 1)	NetLogo or NetLogoLab project (part 2 – creating a model)	Wilensky, U. (2001). *Modeling Nature's Emergent Patterns with Multi-agent Languages.* Wilensky, U. & Resnick, M. (1999). *Thinking in Levels…* Martin et al (2000). *To Mindstorms and Beyond…*
February 16 Class 6 – February 17	Readings Response, 5pm NetLogo/NetLogoLab project (part 2)	Software Review Final project design specification	Blikstein, P., & Wilensky, U. (2007). *Bifocal modeling…* Wilensky, U., & Reisman, K. (2006). *Thinking Like a Wolf, a Sheep…*
February 23 Class 7 - February 24	Readings Response, 5pm **Final Project Design Specification** Software Review		Perkins, D. (1991). *Technology Meets Constructivism: Do They Make a Marriage?* Eisenberg, M. (2003). *Mindstuff: Educational Technology Beyond the Computer.*
March 2 Class 8 – March 3	Readings Response, 5pm Final Project Functional Specification (or) Final Project Curricular Specification		DiSessa, A. (2000). *Changing Minds* (Introduction and Chapter 1)
March 9 Class 9 - March 10 March 16	Readings Response, 5pm Final Project		TBD
Class 10 - March 17	Final Project Presentations		

Course readings

Abrahamson, D., Berland, M.W., Shapiro, R. B., Unterman, J. W., & Wilensky, U. (2004). Leveraging epistemological diversity through computer-based argumentation in the domain of probability. In Y. B. Kafai, W. A. Sandoval, N. Enyedy, A. S. Nixon, F. Herrera (Eds.), *Proceedings of The Sixth International Conference of the Learning Sciences* (pp. 28 – 35). Mahwah NJ: Lawrence Erlbaum Associates.

Ackermann, E. K. (2001). *Piaget's constructivism, Papert's constructionism: What's the difference?* Future of learning group publication.

Blikstein, P., & Wilensky, U. (2007). *Bifocal modeling: a framework for combining computer modeling, robotics and real-world sensing.* Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL, April 9-13.

diSessa, A. (2000). *Changing Minds: Computers, Learning, and Literacy*. Cambridge: MIT Press. (Introduction and Chapter 1)

diSessa, A. A. (1997). Open toolsets: New ends and new means in learning mathematics and science with computers. In E. Pehkonen (Ed.), *Proceedings of the 21st Conference of the International Group for the Psychology of Mathematics Education*, Vol. 1, Lahti, Finland, 47 – 62.

Edwards, L. (1995). Microworlds as Representations. In A. diSessa, C. Hoyles, and R. Noss (Eds.), *Computers and Exploratory Learning.* NATO ASI Series, Subseries F, 146. Heidelberg: Springer-Verlag.

Eisenberg, M. (1991). *Programmable Applications: Interpreter Meets Interface.* MIT AI Lab Memo.

Eisenberg, M. (2003). *Mindstuff: Educational Technology Beyond the Computer*. Paper based on talk at the University of Colorado-Boulder's Institute for Cognitive Science, December 2003.

Falbel, A. (1991). The Computer as a Convivial Tool. In I. Harel & S. Papert (Eds.). *Constructionism.* (p. 29 – 40). Norwood, New Jersey: Ablex Publishing

Goldman-Segall, R. & Maxwell, J.W. (2002). Computers, the Internet, and new media for learning. In W. M. Reynolds & G. E. Miller (Eds.), *Handbook of psychology. Volume 7: Educational psychology* (pp 393–427). New York: John Wiley & Sons.

Hancock, C. (2001). Children’s Understanding of Processes in the Construction of Robot Behaviors.

Harel, I., and Papert, S. (1990). Software Design as a Learning Environment. *Interactive Learning Environments*, vol. 1, no. 1, pp. 1-32.

Kay, A. (1991). Computers, Networks, and Education. *Scientific American,* vol. 265, no. 3, pp. 100-107 (Sept. 1991).

Martin et al (2000). To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines. In *Robots for Kids: Exploring New Technologies for Learning Experiences.* (Edited by Allison Druin). Morgan Kaufman / Academic Press, San Francisco,

Oren, T. (1990). Designing a New Medium. In *The Art of Human-Computer Interface Design* (edited by B. Laurel). Reading, MA: Addison Wesley.

Papert, S. (1991). Situating Constructionism. In *Constructionism*, edited by I. Harel and S. Papert. Norwood, NJ: Ablex Publishing.

Perkins, D. (1991). Technology Meets Constructivism: Do They Make a Marriage? *Educational Technology*, May 1991.

Piaget, J.(1952). Conservation of Continuous Quantities. *The child's conception of number,* (p. 3-17). London: Routledge and Kegan.

Piaget, J. (1929). *The child's conception of the world.* (p. 194-206) London/New York: Harcourt, Brace, and World.

Piaget, J. (1952). *The Origins of Intelligence in Children.* (p. ix – 20) New York, NY: International University Press.

Resnick, M., Martin, F., Sargent, R. & Silverman, B. (1996). Programmable Bricks: Toys to Think with. *IBM Systems Journal.* Vol. 35, Nos. 3&4.

Smith, D., Cypher, A. & Tesler, L (2000). Novice Programming comes of Age. In H. Lieberman (Ed.), *Your Wish is My Command.* Cambridge, MA: MIT Press.

Starr, P. (1994). Seductions of Sim: Policy as a Simulation Game. *The American Prospect,* 5(17), March 21, 1994.

Turkle, S. (1984). Adolescence and Identity: Finding Yourself in the Machine. *The Second Self: Computers and the Human Spirit.* New York: Simon and Schuster.

Turkle, S., Papert, S. (1991). Epistemological Pluralism and Revaluation of the Concrete. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 161-192). Norwood, NJ: Ablex Publishing Co.

Wilensky, U. (2001) *Modeling Nature's Emergent Patterns with Multi-agent Languages.* Proceedings of EuroLogo 2001. Linz, Austria.

Wilensky, U. & Reisman, K. (2006). Thinking Like a Wolf, a Sheep or a Firefly: Learning Biology through Constructing and Testing Computational Theories -- an Embodied Modeling Approach. *Cognition & Instruction, 24*(2), pp. 171-209.

Wilensky, U. & Resnick, M. (1999). Thinking in Levels: A Dynamic Systems Perspective to Making Sense of the World. *Journal of Science Education and Technology*. Vol. 8 No. 1. pp. 3 – 18.

Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of Knowledge Disciplines through new representational forms. In J. Clayson & I. Kallas (Eds.), Proceedings of the Constructionism 2010 Conference. Paris, France.

The software programs we will be using in class all have accompanying tutorials and manuals embedded in the software:

- Microworlds EX Manuals
- NetLogo Manual
- NetLogo GoGo Board Extension Instructions

Extra Readings (for reference only)

Books

Bolter, J. (1991). *Writing: The Computer, Hypertext, and the History of Writing*. Hillsdale, NJ: Lawrence Erlbaum.

Bowers, C. A. (1988). *The Cultural Dimensions of Educational Computing: Understanding the non-neutrality of Technology.* Teachers College Press.

Geisert, P. & Futrell, M. (1995). *Teachers, Computers and Curriculum* . Allyn & Bacon.

Goody, J. (1987). The interface between the written and the oral. New York: Cambridge University Press.

Grabe, M & Grabe, C. (1996). *Integrating Technology for Meaningful Learning*. Houghton Mifflin.

Harel. I & Papert, S. (Eds.) (1990). *Constructionism*. Norwood, NJ: Ablex.

Harel, I. (1991) *Children Designers: Interdisciplinary Constructions for Learning and Knowing Mathematics in a Computer-Rich School.* Norwood, NJ: Ablex Publishing. ISBN 0-89391-788-5.

Illich, I. (1976). *Tools for Conviviality*. New York: Harper & Row.

Kafai & Resnick (1993) *Constructionsim in Practice*. Mahwah, NJ: Lawrence Erlbaum.

Knapp, L. & Glenn, A. (1996). *Restructuring Schools with Technology.* Allyn & Bacon.

Laurel, B. (Ed.) (1990). The Art of Human Interface Design. New York: Addison Wesley.

Maddux et al (1997). *Educational Computing.*New York: Allyn & Bacon.

Male, N. (1997). *Technology for Inclusion*. New York: Allyn & Bacon

McLuhan, M. (1964). *Understanding Media*. New York: McGraw Hill

Norman, D. A. (1988). *The Psychology of Everyday Things*. New York: Basic Books.

Ong, W. J. (1982). *Orality and Literacy*. London: Routledge.

Papert, S. (1992). *The Children's Machine: Rethinking Schools in the Age of the Computer*. New York: Basic Books.

Papert, S. (1996). *The connected family: Bridging the digital generation gap*. Atlanta, GA: Longstreet.

Perelman, L. (1992). *School is out.* New York: Avon Books.

Perkins, D. et al (Eds.) (1995). *Software Goes to School: Teaching for Understanding with new technologies.* Cambridge: Oxford University Press.

Postman, N. (1992). *Technopoly. The surrender of culture to technology.* New York: Knopf.

Solomon, C. (1986). *Computer Environments for Children.* Cambridge, MA: MIT Press

Taylor, R. (1980). *The Computer in the School: Tutor, Tool, Tutee.* New York: Teachers College Press.

Turkle, S. (1995). *Identity in the age of the internet.* New York: Simon & Schuster.

Articles

Ackerman, E. (1991). From Decontextualized to Situated Knowledge: Revisiting Piaget's Water-Level Experiment. In I. Harel & S. Papert (Eds.) Constructionism. (p. 269 – 294). Norwood, New Jersey: Ablex Publishing.

Ackerman, E. (1996). Perspective-Taking and Object Construction: Two Keys to Learning. In Y. Kafai & M. Resnick (Eds.) Constructionism in Practice (p. 25 – 36). Mahwah: NJ: Lawrence Erlbaum Associates.

Anderson, J.R., Corbett, A., Koedinger, K. & Pelleetier, R. (1995). Cognitive Tutors: Lessons Learned. Journal of the Learning Sciences, 4, 2.

Bamberger, J. (1996). Turning Music Theory on its Ear: Do we hear what we see; Do we see what we say? International Journal of Computers for Mathematical Learning, 1(1), 33-55.

Bareiss, R. & Beckwith, R. (1993). Advise the President: A Hypermedia System for Teaching Contemporary American History. Presented at American Educational Research Association. Atlanta, GA.

Blikstein, P. (2015). Computationally Enhanced Toolkits for Children: Historical Review and a Framework for Future Design1. Interaction, 9(1), 1-68.

Borovoy, R., McDonald, M., Martin, F., Resnick, M. (1996). Things that Blink: Computationally Augmented Name Tags. IBM Systems Journal, 35(3).

Bruckman, A. Programming for Fun: MUDs as a context for collaborative learning. MIT Media Lab.

Bruckman, A. E. (2000). Situated Support for Learning: Storm's Weekend with Rachael.

Journal of the Learning Sciences, 9 (3), 329 – 372.

Buechley, L., Eisenberg, M., Catchen, J. and Crockett, A. (2008). The LilyPad Arduino: Using Computational Textiles to Investigate Engagement, Aesthetics, and Diversity in Computer Science Education. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI), Florence, Italy, April 2008, pp. 423-432.

Collins, A. & Brown, J.S. (1988). The Computer as a Tool for Learning Through Reflection. In H. Mandl & A. Lesgold (Eds.) Learning Issues for Intelligent Tutoring Systems (pp. 1-18). New York: Springer Verlag.

Confrey, J. (1993). The role of technology in reconceptualizing functions and algebra. In J.R. Beceker & B.J. Pence (Eds.), Proceedings of the Fifteenth Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education Vol. 1, Pacific Grove, CA, October 17-20. San Jose, CA: San Jose State University, Center for Mathematics and Computer Science Education, p. 47-74.

diSessa, A. A., Abelson, H., & Ploger, D. (1991). An overview of Boxer. The Journal of Mathematical Behavior, 10(1), 3-15.

diSessa, A.A., Hammer, D., Sherin, B., & Kolpakowski, T. (1991). Inventing graphing: Meta-representational expertise in children. Journal of Mathematical Behavior, 10(2), 117-160.

Duffy, T. & Jonassen, D. (1991). Constructivism: New implications for instructional technology?, Educational Technology, May 1991, 7-12.

Edelson, D., Pea, R., Gomez, L. (1996). Constructivism in the collaboratory. In B. G. Wilson (Ed.), Constructivist learning environments: Case studies in instructional design. Englewood Cliffs, NJ: Educational Technology Publications.

Fischer, G., & Lemke, A. C. (1987). Construction kits and design environments: Steps toward human problem-domain communication. Human-Computer Interaction, 3, 179-222.

Hancock, C., Kaput, J. & Goldsmith, L. (1992). Authentic inquiry with data: Critical Barriers to classroom implementation, Educational Psychologist, 27 (3), 337-364.

Hancock, C. (1993).The Medium and the Curriculum: Reflections on Transparent Tools and Tacit Mathematics. In A. A. diSessa, C. Hoyles, R. Noss (Eds.) Computers and Exploratory Learning (p. 221 - 240). Springer.

Harvey, B. (1991). Symbolic Programming vs. the A.P. Curriculum. The Computing

Teacher, vol. 18, no. 5 (Feb. 1991)

Horwitz, P. Contrasting styles in the design of science software. Bolt Beranek and Newman, Inc.

Horwitz, P., Neumann, E. & Schwartz, J. (1994). The Genscope Project, Connections, Spring 1994, 10-11.

Horwitz, P. (1989). ThinkerTools: Implications for science teaching. In J.D. Ellis (Ed.), 1988 AETS Yearbook: Information technology and Science Education.

Horwitz, P., Taylor, E. & Barowy, W. (1994). Teaching special relativity with a computer, Computers in Physics, 8 (1), 92-97.

Hoyles, C. (1991). Computer-based learning environments for Mathematics. In A. Bishop, S. Mellin-Olson, and J. Van Dormolen (Eds.), Mathematical Knowledge: Its growth Through Teaching. Dordrecht: Kluwer, 147-172.

Kahn, K. (2000). Generalizing from Examples. In Your Wish is My Command. (Edited by Henry Lieberman).

Kafai, Y. & Harel, I. (1991). Learning Through Design and Teaching:Exploring Social and Collaborative Aspects of Constructionism. In I. Harel & S. Papert (Eds.) Constructionism. (p. 85 – 110). Norwood, New Jersey:Ablex Publishing.

Kafai, Y., Feldon, D., Fields, D. A., Giang, M., & Quintero, M. (2007). Life in the time of Whypox: A virtual epidemic as a community event. In C. Steinfield, B. Pentland, M. Ackerman, &. N Contractor (Eds.), Communities and Technologies 2007 (pp. 171-190). New York: Springer.

Kaput, J., Noss, R. & Hoyles, C. (2001). Developing New Notations for a Learnable Mathematics in the Computational Era. http://www.simcalc.umassd.edu/downloads/developingnotations.pdf.

Kay, A., & Goldberg, A. (1977). Personal dynamic media. IEEE Computer, 31-41.

Koedinger, K. R. & Anderson, J. R. (1993). Effective use of intelligent software in high school math classrooms. In Proceedings of the World Conference on Artificial Intelligence in Education, (pp. 241-248). Charlottesville, VA: AACE.

Koschmann, T. (1993). Using Technology to Assist in Realizing Effective Learning and Instruction: A Principled Approach to the Use of Computers in Collaborative Learning. Journal of the Learning Sciences, 3(3).

Kozma, R. B. (1991). Learning with media. Review of Educational Research, 61(2), 179-211.

Kuttner, R. The revival of a lost art, The Boston Globe.

Nemirovsky, R., Tierney, C., Wright, T. (1998). Body Motion and Graphing. Cognition and Instruction, 16(2), 119-172.

Noss, R.& Hoyles, C. (1991). Logo and the Learning of Mathematics: Looking Back and Looking Forward. In Hoyles, C. & Noss, R. (Eds.) Learning Mathematics and Logo. London: MIT Press. p. 431-468.

Papert, S. (1990). A critique of technocentrism in thinking about the school of the future. Epistemology and Learning Group Memo No. 2. MIT Media Laboratory: Cambridge, MA.

Papert, S. (1993). Literacy and Letteracy in the media ages, Wired, May/June 1993.

Papert, S., and Solomon, C. (1971). Twenty Things to do with a Computer. Artificial Intelligence Memo 248, MIT AI Laboratory. Cambridge, MA.

Pea, R., Edelson. D. & Gomez, L.M. (1994). Distributed Collaborative Science Learning Using Scientific Visualization and Wideband telecommunications. Paper presented at the 160th meeting of the American Association for the Advancement of Science.

Reiser, B. (1989). Pedagogical Strategies for Human and Computer Tutoring. Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco, CA, March 30-April2.

Repenning, A. (1994). Programming substrates to create interactive learning environments, Interactive Learning Environments, 4 (1), 45-74.

Resnick, M. (1994). "Behavior Construction Kits". Communications of the ACM.

Resnick, M., and Ocko, S. (1991). LEGO/Logo: Learning Through and About Design. In Constructionism (ed. by I. Harel and S. Papert). Norwood, NJ: Ablex Publishing.

Roschelle, J. (1994). Collaborative inquiry: Reflections on Dewey and learning technology. Computing Teacher, 21(8), 6-8.

Scardamalia, M. & Bereiter, C. (1991). Higher levels of agency for children in knowledge building: A challenge for the design of new knowledge media, The Journal of the Learning Sciences, 1 (1), 37-68.

Schank, R.C. & Jona, M.Y. (1991). Empowering the student: New perspectives on the design of teaching systems, The Journal of the Learning Sciences, 1 (1), 7-35.

Schank, R. (1994). Goal-Based Scenarios: A Radical Look at Education. Journal of the Learning Sciences, Vol. 3 No. 4.

Schwartz, J. (1989). Intellectual Mirrors: A Step in the Direction of Making Schools Knowledge-Making Places. Harvard Educational Review.

Shaffer, D. & Kaput, J. (1999). Mathematics and virtual culture: An evolutionary perspective on technology and mathematics education. Educational Studies in Mathematics, 37(2).

Sherin, B. (2001). A Comparison of Programming Languages and Algebraic Notation as Expressive Languages for Physics. International Journal of Computers for Mathematical Learning, 6(1), 1-61.

Soloway, E., Guzdial, M., & Hay, K. E. (1994). Learning Centered Design. Interactions, 1(2).

Thornton, R. (1993). Changing the Physics Teaching Laboratory: Using technology and new approaches to learning to create an experiential environment for learning physics concepts, Proceedings of the Europhysics Conference on the Role of Experiment in Physics Education. Ljubljana, Slovinia.

Vossoughi, S., Hooper, P. K., & Escudé, M. (2016). Making through the lens of culture and power: Toward transformative visions for educational equity. Harvard Educational Review, 86(2), 206-232.

Wilkerson-Jerde, M., Wagh, A., & Wilensky, U. (2015). Balancing curricular and pedagogical needs in computational construction kits: Lessons from the DeltaTick project. Science Education, 99(3), 465-499.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. Journal of Science Education and Technology, 25(1), 127-147.

White, B., Horwitz, P. (1987). Thinker Tools: Enabling Children to Understand Physical Laws. Cambridge, MA: BBN Laboratories, Inc.

White, B. Y. and Frederiksen, J. R. (2000). Technological Tools and Instructional Approaches for Making Scientific Inquiry Accessible to All. Innovations in Science and Mathematics Education: Advanced Designs for Technologies of Learning. M. J. Jacobson and R. B. Kozma. Mahwah, NJ, Lawrence Erlbaum Associates: 321-360.

Wilensky, U. (2003). Statistical mechanics for secondary school: The GasLab modeling toolkit. International Journal of Computers for Mathematical Learning, 8(1), 1-41 (In special issue on agent-based modeling,, U. Wilensky (Ed.)).

Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of Knowledge Disciplines through new representational forms. In J. Clayson & I. Kallas (Eds.), Proceedings of the Constructionism 2010 Conference. Paris, France.