The Design of Technological Tools for Thinking and Learning

General Information

Learning Sciences 426, 326 / CS 496
Winter 2025, Wednesdays, 2:00-4:50 PM
Annenberg Hall 303

Professor

Uri Wilensky
Tel. (847) 467-7593, Annenberg 337
uri@northwestern.edu

Teaching Assistant

Kritphong Mongkhonvanit
Tel. (224) 382-3515, Annenberg 225
kritphong@u.northwestern.edu
Office hours: Fridays from 2-3 in Annenberg 225 and by appointment

Fill out beginning course survey: By January 6th
https://forms.gle/6cUPN1zcps4qA3XJ7
First class: **January 8th**
Last class: **March 12th**
Final poster presentations: March **19th** (**Note**: The official final exam time for this course would be March 18th 3-5pm. However, it is preferred to do it in a three hour block on the afternoon of March 19th. Please inform us right away if you have a conflict with the afternoon of March 19th).

Course Web Sites:

https://ccl.northwestern.edu/dtttl/2025LS426/Home.html
https://canvas.northwestern.edu/courses/227121

Email Lists
Course Instructors: cd-fac@ccl.northwestern.edu (use this address to ask for TA help)
Most often, we'll send email through the course canvas page.

AI Tool Policy: You are free to use the grammar checking tools such as Grammarly and LanguageTool in this course. The use of any other AI tools (such as ChatGPT, Claude, etc.)

are not allowed by default because we think their use will decrease your learning. If you have a special use case that you think would be beneficial for your learning, send an email to cd-fac@ccl.northwestern.edu explaining your proposed use case and we will decide whether to approve it.

Please note that the specifics of this course syllabus are subject to change in the case of unforeseen circumstances. Instructors will notify students of any changes as soon as possible. Students will be responsible for abiding by the changes.

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://nu-ls426-2025.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. We will be using NetLogo 6.4. Download - 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, Crayon Physics, 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. The course final project is a solo project.

In person attendance is expected, except of course, if you have a positive covid test or illness symptoms.

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.
- 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 19th, 2025).

In addition, due to the group project nature of the class, you are also asked to send an 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 group-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 19th.
- The final project functional specification (or curricular flow specification) is due by March 5th.
- The final project is due by March 15th.
- Final projects will be presented at a public poster session on March 19th. There is food and 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 the due date, 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 19th 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 by the first class:

- 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 4:00 pm on Tuesdays.

Students requesting the use of assistive technology as an accommodation should contact Accessible NU. Some classes may be recorded. Unauthorized use of classroom recordings – including distributing or posting them is prohibited.

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 8		Group Quilt Project	"Getting Started", from the DTTTL website
January 14 Class 2 – January 15	Readings Response, 4pm 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 class – this will be true for all subsequent reading assignments)
January 21 Class 3 – January 22	Readings Response, 4pm Lynx (Logo) Coding (Part 2)		Harel, I., and Papert, S. (1990). *Software Design as a Learning Environment.* Edwards, L. (1995). *Microworlds as Representations.*
January 28 Class 4 - January 29	Readings Response, 4pm Lynx (Logo) Coding interactive media 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 4 Class 5 – February 5	Readings Response, 4pm 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 11 Class 6 – February 12	Readings Response, 4pm NetLogo/NetLogoLab project (part 2)	Software Review Final project design specification	Blikstein, P., & Wilensky, U. (2007). *Bifocal modeling… {This is not a complete title}* Wilensky, U., & Reisman, K. (2006). *Thinking Like a Wolf, a Sheep… {complete the title.*
February 18 Class 7 - February 19	Readings Response, 4pm Draft Project Design Specification Software Review	Revised Final project design specification	Perkins, D. (1991). *Technology Meets Constructivism: Do They Make a Marriage?*
February 25	Readings Response, 4pm	Final Project Functional Specification (or) Final Project Curricular Specification	Eisenberg, M. (2003). *Mindstuff: Educational Technology Beyond the Computer.*

Class 8 – February 26			Buechley & Perner-Wilson (2012). *Crafting Technology: Reimagining the Processes, Materials, and Cultures of Electronics*
March 4 Class 9 - March 5	Readings Response, 4pm Final Project Functional Specification (or) Final Project Curricular Specification Readings Response, 4pm		Roque, R. (2020). Building Relationships and Building Projects: Designing for Family Learning.
March 11 Class 10 - March 12	Readings Response, 4pm Final Project (due March 15th)		DiSessa, A. (2000). *Changing Minds* (Introduction and Chapter 1) Wilensky & Papert (2010). *Restructurations: Reformulating Knowledge Disciplines through New Representational Forms*
Class 11 - March 19 – Presentations and public poster session	Final Project Presentations		

Course readings

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.

Blikstein, P. (2013). Digital Fabrication and 'Making' in Education: The Democratization of Invention. In J. Walter-Herrmann & C. Büching (Eds.), FabLabs: Of Machines, Makers and Inventors. Bielefeld: Transcript Publishers.

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

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

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.

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

Kafai, Y. B., Fields, D. A., Roque, R., Burke, W. Q., & Monroy-Hernandez, A. (2012). Collaborative agency in youth online and offline creative production in Scratch. Research and Practice in Technology Enhanced Learning, 7(2), 63-87.

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.

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

Resnick, M., Maloney, J., Monroy, Hernández, A., Rusk, N., Eastmond, E., Brennan, K., & Silverman, B. (2009). Scratch: Programming for all. Communications of the ACM, 52(11), 60-67.

Shaw, A. (1996). Social Constructionism and the Inner City: Designing Environments for Social Development and Urban Renewal. In (Kafai, Y. & Resnick, M., Eds.) Constructionism in Practice Designing, Thinking, and Learning in A Digital World.

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.

Wilkerson-Jerde, M. H. (2014). Construction, categorization, and consensus: student generated computational artifacts as a context for disciplinary reflection. Educational Technology Research & Development, 62(1), 99-121.

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

- Lynx Manuals
- NetLogo 6.4.0 Manual
- NetLogo GoGo Board Extension Instructions

Extra Readings (for reference only)

Books for reference

Bers, M. (2021). Coding as a Playground (2nd ed.)

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.

Freire, P. (1970). Pedagogy of the oppressed. New York: Herder & Herder.

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

Goldenberg, E. P., & Feurzeig, W. (1987). *Exploring language with Logo*. Mit Press.

Moody, 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.

Holbert, N. Berland, M. & Kafai, Y. (Eds.) (2020), Designing constructionist futures: The art, theory, and practice of learning designs. Cambridge, MA: MIT Press. pp. 387-390.

Hooks, B. (1994). Teaching to transgress: education as the practice of freedom. New York: Routledge. 1994. ISBN 978-0-415-90808-5.

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

.

Kafai & Resnick (1993) *Constructionism 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.

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

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

Articles for reference

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.

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.

Ackermann, E. (2004). Constructing knowledge and transforming the world. In M. Tokoro & L. Steels (Eds.), A learning zone of one's own: Sharing representations and flow in collaborative learning environments (pp. 15-37). Amsterdam: IOS Press.

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

Ball, M., Monig, J., Romagoisa, B & Harvey, B. (2019). Snap! A Look at 5 Years, 250,000 Users and 2 Million Projects. SIGCSE '19: Proceedings of the 50th ACM Technical Symposium on Computer Science Education. February 2019, page 1279. https://doi.org/10.1145/3287324.3293863.

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.

Berland, M. (2016). Making, Tinkering, and Computational Literacy. In Makeology Makers as Learners (Eds. Peppler, K. Halverson, E. & Kafai, Y.). https://doi.org/10.4324/9781315726496.

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

Blikstein, P. (2008). Travels in Troy with Freire: Technology as an agent for emancipation. In In P. Noguera & C. A. Torres (Eds.) (Ed.), Social justice education for teachers: Paulo Freire and the possible dream (pp. pp. 205–244). Rotterdam, Netherlands: Sense.

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

Bruckman, A. (1994). *Programming for Fun: MUDs as a Context for Collaborative Learning.* Paper

presented at the National Educational Computing Conference, Boston, MA, June 1994.

Bruckman, A. E. (2000). Situated Support for Learning: Storm's Weekend with Rachael. Journal of the Learning Sciences, 9 (3), 329 – 372.

Bruckman, A. (1998). Community Support for Constructionist Learning. Computer Supported Cooperative Work (CSCW) 7, 47–86 (1998). https://doi.org/10.1023/A:1008684120893.

Buechley, L., & Eisenberg, M. (2006). Electronic/Copmputational textiles and children's crafts. In Proceedings of Interaction Design and Children (IDC 2006) (pp. 49-56). Tampere, Finland.

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. Becker & 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.

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.

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.

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

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

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

Guo, Y., Wagh, A., Brady, C., Levy, S. T., Horn, M. S., & Wilensky, U. (2016). Frogs to Think with: Improving Students' Computational Thinking and Understanding of Evolution in A Code-First Learning Environment. Proceedings of the 15th International Conference of ACM SIGCHI Interaction Design and Children (IDC 2016).

Guo, Y. & Wilensky, U. (2018). Mind the Gap: Teaching high school students about wealth inequality through agent-based participatory simulations. In Dagiene V. & Jasute E. (Eds.), *Proceedings of the Constructionism 2018* conference (pp. 238-250). Vilnius, Lithuania.

Hancock, C. (2001). Children's understanding of process in construction of robot behaviors. Paper presented at the Annual Meeting of the American Educational Research Association, Seattle, WA.

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).

Hjorth, A. (2018). Social Gears - Constructionism and Social Studies. Keynote paper. In Dagiene V. & Jasute E. (Eds.), Proceedings of Constructionism 2018, Vilnius,

Lithuania.

Horn, M., Banerjee, A., West, M., Pinkard, N., Pratt, A., Freeman, J., Magerko, B., & McKlin, T. (2020). TunePad: Engaging learners at the intersection of music and code. Proceedings of the International Conference of the Learning Sciences.

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.

Levy, S.T. & Wilensky, U. (2008). Inventing a "mid-level" to make ends meet: Reasoning through the levels of complexity. Cognition & Instruction. 26(1), 1-47.

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.

Sengupta, P., & Shanahan, M. C. (2017). Boundary Play and Pivots in Public Computation: New Directions in STEM Education. International Journal of Engineering Education, 33(3), 1124-1134.

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.

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.

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

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

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.

Weintrop, D. & Wilensky, U. (2018). How Block-based, Text-based, and Hybrid Block/Text Modalities Shape Novice Programming Practices. International Journal of Child-Computer Interaction, 17, 83–92. https://doi.org/10.1016/j.ijcci.2018.04.005

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., Brady, C., & Horn, M. (2014). Fostering Computational Literacy in Science Classrooms. Communications of the ACM, 57(8), 24-28.

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.