INTERNATIONAL CONFERENCE

Constructionism 2018

Constructionism, Computational Thinking
and Educational Innovation

Book of Abstracts

with the Conference Program

August 20-25, 2018 Vilnius, Lithuania

Edited by
Valentina Dagienė and **Eglė Jasutė**

INTERNATIONAL CONFERENCE

Constructionism 2018

Constructionism, Computational Thinking
and Educational Innovation

Book of Abstracts

August 20-25, Vilnius, Lithuania

Organizers:
Vilnius University
Faculty of Philosophy and
Institute of Data Science and Digital Technologies
in cooperation with Lithuanian Computer Society

Edited by
Valentina Dagienė and Eglė Jasutė

ISBN 978-609-95760-2-2

The conference is partially supported by Research Council of Lithuania

CONFERENCE VENUE

Faculty of Philosophy
Vilnius University
Universiteto str. 9
01513 Vilnius, Lithuania
Located at the Old Campus of Vilnius University

The conference website: *http://www.constructionism2018.fsf.vu.lt*

Book of abstracts is available at: *http://www.constructionism2018.fsf.vu.lt/book-of-abstracts*

Full papers are available at: *http://www.constructionism2018.fsf.vu.lt/proceedings*

CONSTRUCTIONISM 2018 COMMITTEES

Conference Chairs

Valentina Dagienė, Vilnius University Institute of Data Science and Digital Technologies, Lithuania
Arūnas Poviliūnas, Vilnius University Faculty of Philosophy, Lithuania
Arnan (Roger) Sipitakiat, Chiang Mai University, Thailand

Steering Committee

Tim Bell, University of Canterbury, New Zealand
Paulo Blikstein, Stanford University, USA
James Clayson, American University of Paris, France
Michael Eisenberg, University of Washington, USA
Gerald Futschek, Vienna University of Technology, Austria
Paul Goldenberg, Education Development Center, USA
Brian Harvey, University of California, USA
Arthur Hjorth, Northwestern University, USA
Celia Hoyles, University College London Institute of Education, UK
Ivan Kalaš, Comenius University, Slovakia
Chronis Kynigos, National and Kapodistrian University of Athens, Greece
Jens Mönig, SAP, Germany
Richard Noss, University College London Institute of Education, UK
Mitchel Resnick, MIT Media Lab, Massachusetts Institute of Technology, USA
Ana Isabel Sacristán, Cinvestav, Mexico
Jenny Sendova, Institute of Mathematics, Bulgaria
Wolfgang Slany, Graz University of Technology Institute of Software Technology, Austria
Gary Stager, Constructing Modern Knowledge, Torrance, USA
Uri Wilensky, Northwestern University, Evanston, USA

Programme Committee

Pavel Boytchev, Bulgaria
Secundino Correia, Portugal
Barbara Demo, Italy
Carina Girvan, UK
Paul Goldenberg, USA
Eric Klopfer, USA
Witold Kranas, Poland
Manolis Mavrikis, UK
Mattia Monga, Italy
Erich Neuwirth, Austria
Mareen Przybylla, Germany
Mitchel Resnick, USA
Ralf Romeike, Germany
Giovanni Serafini, Switzerland
Gary Stager, USA
Eliza Stefanova, Bulgaria
Maciej Syslo, Poland
Márta Turcsányi-Szabó, Hungary
José Armando Valente, Brazil
Jiří Vaníček, Czechia
Michael Weigend, Germany
Michelle Wilkerson, US

Local Organizing Committee

Vladimiras Dolgopolovas
Vida Jakutienė
Eglė Jasutė
Tatjana Jevsikova
Anita Juškevičienė
Aldona Mačiūnienė
Saulius Maskeliūnas
Gabrielė Stupurienė
Olga Suprun

Photo E. Kurauskas

Photo E. Kurauskas

Welcome Message

Dear Colleagues,

We would to like to welcome you to the Constructionism conference.

The Constructionism conference celebrates its fifth anniversary under this name, building on the 27-year tradition of biennial *Eurologo* conferences established by the European Logo community. Logo is the computer programming language for learners in which the constructionist approach was first developed. Seymour Papert, who has coined the term *constructionism,* in order to define its meaning, started from the comparison with the term *constructivism*:

> **Constructionism shares constructivism's connotation of learning as 'building knowledge structures' irrespective of the circumstances of the learning. It then adds the idea that this happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe"** (Papert, S. & Harel, I. Constructionism. New York: Ablex Publ. Corp., 1991).

The constructionism shares the main idea of genetic epistemology elaborated by Jean Piaget about the immanence of the cognitive development. In Piaget's version, the immanent algorithm of the cognitive development includes the sensorimotor, preoperational, concrete operational, and formal operational stages.

The pioneer of the radical constructivism Ernst von Glasersfeld has enriched the constructivist educational discourse by introducing ideas from the Italian philosopher Giambattista Vico. In his Magnum opus *Scienza Nuova*, Vico has elaborated the principle *verum esse ipsum factum* ("What is true is precisely what is made"). According to Vico, "/…/ the world of civil society has certainly been made by men, and that its principles are therefore to be found within the modifications of our own human mind." The world of civil society differs from the world of nature, "which, since God made it, He alone knows" (Vico, J.-B. The New Science. Transl. by T. G. Bergin & M. H. Fisch. Ithaca, N Y: Cornell Univ. Press, 1948 [1744]: 85).

Another important idea brought from Vico is the application of tropology for the description of different modes of consciousness or phases of its development. American post-structuralist Hayden White has applied tropes for the explanation of different phases of cognitive development discovered by Piaget. It seems that it is important for the development of constructionism to track the processes in the field of constructivism and vice versa.

Seymour Papert has stressed that the picture of how scientists actually work should be shared with children but not in the way of verbally-expressed formal knowledge. Closer cooperation with the Science and Technology Studies

could assist in identifying more adequate forms how to communicate scientific knowledge and how to engage children into the attractive and actual scientific research.

What have we achieved until today? Has our concept of education in general and computer science (computing or informatics) education in particular changed and its quality improved? Can we use the lessons of the past to prepare for the future? In an increasingly interdependent and complex world, how is technology and informatics changing society and affecting education through the subject areas of humanities, science, mathematics, and arts? The Constructionism conference has addressed these questions by offering experts from across the world the opportunity to exchange ideas and knowledge, and to generate a more informed understanding of the issues of informatics and digital technologies in education. The Conference offered a number of themes:

- Constructionist epistemology
- Informal learning
- Innovative computing education
- Learning from pioneers
- Methodologies, tools and technologies
- Monitoring, evaluation and research
- Outreach and communication
- Progressive education in national and regional contexts
- Social justice, equity and citizenship
- Teacher training and educational policies
- Technology unplugged activities
- Visual and creative arts

The Conference brings together delegates from all over the world to address pressing issues in computing education. In addition to keynote speakers, research and practice papers, panels, posters, demonstrations, and workshops, the Conference provides facilities and exposure for working groups for the first time. The working groups are formed by participants with a common interest in a topic. Participating in a working group provides a unique opportunity to work with people from different countries who are interested and knowledgeable in the area of the working group. It is also one of the best ways to become part of the constructivist community. Seven working groups have been accepted, covering a broad spectrum of topics. Participants present their preliminary results to conference attendees at a special working group presentation session, and submit a final report after the conference concludes.

The Constructionism conference continues to be truly international with about 150 submissions from 40 countries. The accepted submissions consisted of 18 keynotes talks, 57 research and practice papers, 3 panels, 7 working group proposals, and 27 proposals for posters, demonstrations and workshops. In addition, a special Teachers' Day is organized before the conference: 13 workshops were proposed for more than 150 Lithuanian teachers.

Selected research papers will be published in the international peer-reviewed journals "Informatics in Education**"** and "Problemos**".** After the conference, there will be possibility to extend the best papers and publish them in the international peer-reviewed journal "Constructivist Foundations**".**

We are grateful to all committee members participating in any way in the conference.

Welcome to Vilnius, the capital of Lithuania! Enjoy the Constructionism conference in the country celebrating 100 years of independence!

Conference Co-Chairs: *Valentina Dagienė*
Arūnas Poviliūnas

PROGRAM AT A GLANCE

	Monday, **August 20**	Tuesday **August 21** Location: Main Building, The Theatre Hall*	**Time**	Wednesday **August 22** Location: The Faculty of Philosophy**	Thursday **August 23** Location: The Faculty of Philosophy**	Friday **August 24** Location: The Faculty of Philosophy**	Saturday **August 25** Location: The Faculty of Philosophy**
TEACHERS DAY	9^{00}–10^{00} Registration (in front of the Theatre Hall)	8^{30}–all day Registration	8^{30}–10^{30}	In parallel: • Plenary session III • Plenary session IV	In parallel: • Plenary session V • Plenary session VI	In parallel: • Plenary session VII • Plenary session VIII	In parallel: • Plenary session IX • Plenary session X
	10^{00}–12^{00} Welcome Speakers *(Main Building, The Theatre Hall*)*	10^{00}–11^{30} Excursion to Old Vilnius University I	10^{30}–11^{00}	In parallel: • Working Group presentations I • Panel discussion II	In parallel: • Poster session I • Poster session II	In parallel: • Paper session 9 • Demo session 1	In parallel: • Demo session 2 • Demo session 3 • Workshop 6
			11^{00}–11^{30}	Coffee break	Coffee break	Coffee break	11^{30}–12^{00} Closing: Farewell buffet *Location: Grand Courtyard next to the Main Building**
	12^{00}–13^{00} Lunch (Registration in the Faculty of Philosophy**)	11^{30}–13^{00} Excursion to Old Vilnius University II	11^{30}–13^{30}	In parallel: • Paper session 1 • Paper session 2 • Workshop 1 • Workshop 2	In parallel: • Paper session 6 • Paper session 7 • Paper session 8 • Panel discussion III	In parallel: • Paper session 10 • Paper session 11 • Paper session 12	
	13^{00}–15^{00} Workshops in parallel (The Faculty of Philosophy**)		13^{30}–14^{30}	Lunch	Lunch	Lunch	13^{00}–20^{00} Post Conference Excursions (not included in conference fee)
	15^{00}–15^{30} Coffee Break	14^{00}–16^{00} Plenary session I	14^{30}–16^{00}	In parallel: • Working Group presentations II • Working Group presentations III	14^{30}–22^{00} Excursion Dinner	In parallel: • Paper session 13 • Paper session 14 • Workshop 3	
	15^{30}–17^{00} Workshops in parallel (The Faculty of Philosophy**)	Coffee break	16^{00}–16^{30}	Coffee break		Coffee break	
	17^{00}–18^{00} Reflections and Panel Discussion (The Faculty of Philosophy**, room 301)	16^{30}–17^{30} Plenary Session II 17^{30}–18^{00} Panel Discussion I	16^{30}–18^{30}	In parallel: • Paper session 3 • Paper session 4 • Paper session 5		In parallel: • Paper session 15 • Paper session 16 • Workshop 4 • Workshop 5	
		18^{00}–19^{30} Welcome Reception					

* Main Building, The Theatre Hall: Universiteto St.3

**The Faculty of Philosophy: Universiteto St. 9

CONFERENCE PROGRAM

Monday, August 20	**Teachers' day (Pre-Conference)**
9^{00}–10^{00}	Registration & Coffee (in front of the Theatre Hall**)**
10^{00}–12^{00}	**Opening Session**: Main Building, The Theatre Hall (Universiteto St. 3) **Welcome** **Giedrius Vaidelis (Lithuania).** *Updating Educational Content: Challenges and Possibilities* **Evgenia Sendova (Bulgaria).** *The Beauty in Science and the Science in Beauty* **Rimantas Želvys (Lithuania).** *Future Education: New Challenges for Lithuania?*
12^{00}–13^{00}	Lunch Registration in the Faculty of Philosophy (Universiteto St. 9)
13^{00}–15^{00}	**Workshops (WS) in parallel:** The Faculty of Philosophy (Universiteto St. 9) **WS 1**: room 106. **Judith Bell** (New Zealand). *Dynamic Teaching Ideas for Teaching Music Theory*. Target audience: primary school teachers & music teachers **WS 3**: room 204. **Paul Goldenberg, Cynthia J. Carter** (USA). *Developing Algebraic Habits of Mind in Students*. Target audience: mathematics teachers for students ages 11–18 **WS 7**: room 111. **Mikko-Jussi Laakso** (Finland). *ViLLE – Electronic Learning Path for Mathematics and Programming*. Target audience: primary school teachers **WS 8**: room 201. **Evgenia Sendova, Nikolina Nikolova** (Bulgaria). *Constructionism in Action: Do we Need to Start from Scratch?* Target audience: all teachers **WS 9**: room 313. **Gary Stager** (USA). *Teaching Coding and Physical Computing*. Target audience: all teachers **WS 10**: room 212. **Jacqueline Staub** (Switzerland). *The Essence of Programming at School – Logo in a Spiral Curriculum*. Target audience: primary and lower secondary school teachers **WS 11**: room 308. **Carol Sperry Suziedelis** (USA). *How to Create and Sustain a Progressive Pedagogy in a Traditional Setting*. Target audience: all teachers **WS 13**: room 306. **Annalise Duca, Angele Giuliano** (Malta), **Sofia Nikitopoulou, Nikoleta Yiannoutsou, Chronis Kynigos** (Greece). *The ER4STEM Repository for Educational Robotics*. Target audience: all teachers
15^{00}–15^{30}	Coffee break
15^{30}–17^{00}	**Workshops (WS) in parallel:** The Faculty of Philosophy (Universiteto St. 9). **WS 2**: room 106. **Tim Bell** (New Zealand). *Computer Science Unplugged for Teachers*. Target audience: primary school teachers **WS 4**: room 204. **Paul Goldenberg, Cynthia J. Carter** (USA). *Puzzles & Programming to Develop Mathematical Habits of Mind in 6–10-year Olds*. Target audience: primary school teachers for students ages: 6–10 **WS 5**: room 201. **Ivan Kalaš** (Slovakia). *Powerful Ideas in Lower Primary Programming: High Time to Recognize Them*. Target audience: educators interested in lower primary computing (pupils aged 5 to 9) and general primary teachers **WS 6**: room 107. **Witek Kranas** (Poland). *SNAP! - Beauty & Joy of Computing (visually)*. Target audience: informatics teachers, lower and upper secondary schools (6-12 grades) **WS 7**: room 111. **Mikko-Jussi Laakso** (Finland). *ViLLE – Electronic Learning Path for Mathematics and Programming*. Target audience: primary school teachers **WS 8**: room 401. **Evgenia Sendova, Nikolina Nikolova** (Bulgaria). *Constructionism in Action: Do we Need to Start from Scratch?* Target audience: all teachers **WS 9**: room 313. **Gary Stager** (USA). *Teaching Coding and Physical Computing*. Target audience: all teachers **WS 10**: room 112. **Jacqueline Staub** (Switzerland). *The Essence of Programming at School – Logo in a Spiral Curriculum*. Target audience: primary and lower secondary school teachers **WS 11**: room 308. **Carol Sperry Suziedelis** (USA). *How to Create and Sustain a Progressive Pedagogy in a Traditional Setting*. Target audience: all teachers **WS 12**: room 205. **Khayriah Massarwe, Igor Verner, Daoud Bshouty** (Israel). *Joyful Learning of Geometry in Cultural Context. Analysis and Construction of Geometric Ornaments*. Target audience: all teachers **WS 13**: room 306. **Annalise Duca, Angele Giuliano** (Malta), **Sofia Nikitopoulou, Nikoleta Yiannoutsou, Chronis Kynigos** (Greece). *The ER4STEM Repository for Educational Robotics*. Target audience: all teachers
17^{00}–18^{00}	Reflections and Panel Discussion: room 301

Tuesday, August 21 / Location: Main Building, The Theatre Hall (Universiteto St. 3)

8^{30}–all day	Registration
10^{00}–11^{30}	Excursion to Old Vilnius University I
11^{30}–13^{00}	Excursion to Old Vilnius University II
14^{00}–16^{00}	Session chair: **Valentina Dagienė** **Opening** **Plenary session I** **Rimantas Želvys.** One Hundred Years of Educational Development in Lithuania **James Clayson.** *Look Closely, Watch What Happens: Visual Modelling and Constructionism*
16^{00}–16^{30}	Coffee break
16^{30}–17^{30}	Session chair: **Arūnas Poviliūnas** **Plenary session II** **Gary Stager.** *Making Constructionism Great Again*
17^{30}–18^{30}	**Panel Discussion I** *Inside the Trojan Horse – A Discussion Among the Next Generation of Constructionists* Sylvia Martinez (moderator), Gary Stager, Amy Dugré, Angela Sofia Lombardo, Susana Tesconi, Tracy Rudzitis, Brian C. Smith, Jaymes Dec
18^{30}–19^{30}	*Welcome Reception / Location: Grand Courtyard*

Wednesday, August 22 / Location: The Faculty of Philosophy (Universiteto St. 9)

8^{30}–10^{30}	Session chair: **Gerald Futschek** **Plenary session III:** room 301 **Carol Sperry Suziedelis.** The *Evolution of a Constructionist Teacher (with Reminders from Seymour Papert)* **Evgenia Sendova**. *Back 100 000(2)*	Session chair: **Chronis Kynigos** **Plenary session IV:** room 302 **Celia Hoyles, Richard Noss**. *Scratchmaths: A Positive Outcome for Constructionism at Scale* **Ivan Kalaš.** *Programming in Lower Primary Years: Design Principles and Powerful Ideas*
10^{30}–11^{00}	**Working Group (WG) presentations I:** room 301 WG 2: **Don Passey,** Loice Victorine Atieno, Wilfried Baumann, Valentina Dagienė. *Developing Constructionism, or a New Learning Concept, Across the Ages.*	**Panel discussion II** room 302 *Constructionism at Scale.* Nathan Holbert (moderator), Matthew Berland, Yasmin Kafai, Richard Noss, Celia Hoyles, Kylie Peppler, Debbie Fields
11^{00}–11^{30}	Coffee break	

11^{30}–13^{30}	Session chair: **Natasa Grgurina** **Paper session 1:** room 301 **Education and innovations** **Arthur Hjorth, Corey Brady, Uri Wilensky.** *Sharing is Caring in the Commons – Students' Conceptions about Sharing and Sustainability in Social-Ecological Systems* **Arthur Hjorth, Uri Wilensky.** *Urban Planning-in-Pieces: A Computational Approach to Understanding Conceptual Change and Causal Reasoning about Urban Planning* **Sugat Dabholkar, Gabriella Anton, Uri Wilensky.** *Developing Mathetic Content Knowledge Using an Emergent Systems Microworld* **Elmara Pereira de Souza, Luísa Moura.** *Constructionism as an Epistemological Option in Courses of Youth Center for Science and Culture – Bahia – Brazil*	Session chair: **Evgenia Sendova** **Paper session 2:** room 302 **Constructionism in Mathematics** **Chantal Buteau, Ana Isabel Sacristán, Eric Muller.** *Teaching in a Sustained Post-Secondary Constructionist Implementation of Computational Thinking for Mathematics* **Maite Mascaró, Ana Isabel Sacristán**. *Assessing Learning through Exploratory Projects in Constructionist R-based Statistics Courses for Environmental Science Students* **Christina Todorova, Carina Girvan, Nikoleta Yiannoutsou, Marianthi Grizioti, Ivaylo Gueorguiev, Pavel Varbanov, George Sharkov.** *Visualizing Mathematics with the MathBot: A Constructionist Activity to Explore Mathematical Concepts through Robotics* **Einari Kurvinen, Valentina Dagiene, Mikko-Jussi Laakso.** *The Impact and Effectiveness of Technology Enhanced Mathematics Learning*	Session chair: **Eglė Jasutė** **Workshop 1:** room 111 **Jacqueline Staub.** *The Essence of Programming at School – Learning for Life* **Workshop 2:** room 111 **Stephen Howell, Lizbeth Goodman.** *Developing Body Tracking Software with Scratch and Kinect*
13^{30}–14^{30}	Lunch		
14^{30}–16^{00}	Session chair: **Gabrielė Stupurienė** **Working Group (WG) presentations II:** room 301 WG 1: **Gerald Futschek, Bernhard Standl,** Chantal Buteau, Andrew Csizmadia, Lilia Georgieva, Lina Vinikienė, Jane Waite. *Constructionist Approaches to Computational Thinking*.	Session chair: **Tatjana Jevsikova** ***Working Group (WG) presentations* III:** room 302 WG 3: **Evgenia Sendova,** Christos Chytas, Katarzyna Olędzka, Ralf Romeike, Wolfgang Slany**.** *Creating and Looking at Art with Logo Eyes*.	

	WG 5: **Michael Weigend,** Kazunari Ito, Anita Juškevičienė, Igor Pesek, Zsuzsa Pluhár, Jiří Vaníček. *Constructionism in the Classroom: Creative Learning Activities on Computational Thinking*. WG 6*:* **Mattia Monga,** Michael Lodi, Dario Malchiodi, Anna Morpurgo, Oluwakemi Oduwole, Bamidele Oluchi, Bernadette Spieler. *Learning to Program in a Constructionist Way.*	WG 4: **Lilija Duoblienė, Jūratė Baranova,** Luc Anckaert, Wilfried Baumann **.***The Constructive Strategies in Teaching Humanities with Films*. WG 7: **Ana Isabel Sacristán,** Richard Akrofi Kwabena Baafi, Lina Kaminskienė, Michael Sabin. *Constructionism in Upper Secondary and Tertiary Levels*.
16^{00}–16^{30}	Coffee break	
16^{30}–18^{30}	Session chair: **Mattia Monga** **Paper session 3:** room 301 **Computational Thinking** **Judith Bell, Tim Bell.** *Computational Thinking and Music Learning* **Marianthi Grizioti, Chronis Kynigos.** *Programming Approaches to Computational Thinking: Integrating Turtle Geometry, Dynamic Manipulation and 3D Space* **Marianthi Grizioti, Chronis Kynigos.** *Constructionist Approaches to Computational Thinking: A Case of Game Modding with ChoiCo* **Anita Juškevičienė, Valentina Dagienė.** *Interconnection Between Computational Thinking and Digital Competence*	Session chair: **Jiří Vaníček** **Paper session 4:** room 302 **Constructionist approaches** **Valentina Dagienė, Gabrielė Stupurienė.** *Short Tasks – Big Ideas: Constructive Approach for Learning and Teaching of Informatics Concepts in Primary Education.* **Miroslava Černochová, Radek Čuma.** *Forming Concepts for Programming Conditional Statements in the Primary School* **Jean Griffin.** *Constructionism and De-Constructionism as Complementary Pedagogies* **Tilman Michaeli, Stefan Seegerer, Ralf Romeike.** *Enabling Collaboration and Tinkering: A Version Control System for Block-based Languages* **Jake Rowan Byrne, Kevin Sullivan, Katriona O'sullivan.** *Active Learning of Computer Science Using a Hackathon-like Pedagogical Model*	Session chair: **Arthur Hjorth** **Paper session 5:** room 306 **Reflections** **Nicolas Pope, Jonathan Foss, Meurig Beynon.** *Reconstructing Constructionism by Construal* **Deborah Fields, Mia Shaw, Yasmin Kafai.** *Personal Learning Journeys: Reflective Portfolios as "Objects-to-Learn-With" in an E-textiles High School Class* **Evgeny Patarakin.** *Using Agent-based Modelling of Collaboration for Social Reflection* **Francesca Agatolio, Alfredo Asiain, Alfredo Pina, Gabriel Rubio, Michele Moro.** *Constructive and Collaborative Digital Storytelling for Enhancing Creativity and Cooperation In and Out of School*
Thursday, August 23 / Location: The Faculty of Philosophy (Universiteto St. 9)		
8^{30}–10^{30}	Session chair: **Gary Stager** **Plenary session V:** room 301 **Gerald Futschek.** *Computational Thinking and Creativity* **Tim Bell.** *CS Unplugged and Computational thinking*	Session chair: **James Clayson** **Plenary session VI:** room 302 **Uri Wilensky.** *Reempowering powerful ideas* **Paulo Blikstein.** *Constructionism Won, Now What? The Role of Constructionist Research in the Age of Ubiquitous Computing*
10^{30}–11^{00}	Session chair: **Anita Juškevičienė** **Poster Session I:** room 301 **Nalin Tutiyaphuengprasert.** *Applied Constructionism: Critical Reflection and Learning Through Play in Adult Learning* **Sawaros Thanapornsangsuth, Nathan Holbert, Monica Chan.** *Towards Girls' Self-perception in Technology and Craft: Challenges and Implications* **Enric Ortega Torres, Vincent Sanjosé López, Joan-Josep Solaz Portolés.** *Influence of Students' Self-perceived Use of Metacognitive Strategies and Sensory Preferences on Academic Achievement in Science and Technology* **Takeshi Watanabe, Yuriko Nakayama, Yasunori Harada, Yasushi Kuno.** *Programming Lessons for Kindergarten Children in Japan* **Sayaka Tohyama, Yugo Takeuchi.** *Collaborative Creative Music Activity with ICT: A Case Study for Children in Grade Five* **Yoshiaki Matsuzawa, Misako Noguchi, Issei Nakano.** *Exploration of Algorithm Abstraction Process with Cubetto and Middle Grade Elementary Kids* **Aoi Yoshida, Kazunari Ito, Kazuhiro Abee.** *A Practical Report on a Programming Course with "Making" Using micro:bit* **Liudmyla Kryvoruchka**. *Heuristic Potential of Open Institutional Models in Researchers Education.*	Session chair: **Lina Vinikienė** **Poster Session II:** room 302 **Michael Tan.** *Constructing what? Knowledge of the powerful, and powerful knowledge* **Carina Girvan, Wilfried Lepuschitz, Ivaylo Gueorguiev, Christina Todorova, Chronis Kynigos, Marianthi Grizioti, Angele Giuliano, Annalise Duca, Julian M. Angel-Fernandez, Markus Vincze.** *Educational Robotics for STEM: From Workshops to Curricula and Framework* **Ivaylo Gueorguiev, Christina Todorova, Nikoleta Yiannoutsou, Xristina Greka, Pavel Varbanov, George Sharkov, Carina Girvan, Julian M. Angel-Fernandez, Lisa Vittori, Annalise Duca.** *Towards a Generic Curriculum for Educational Robotics in STEM: From Scientific Concepts to Technologies and Powerful Ideas* **Barbara Sabitzer.** *Modeling Across the Subjects* **Jinbao Zhang.** *An Experimental Exploration of the Development of Design Thinking in University Maker Courses* **Márton Visnovitz, Győző Horváth.** *The Web – A Platform for Creation* **Pekka Mäkiaho, Timo Poranen, Katriina Vartiainen.** *Construction of a Project Monitoring Application Iteratively and Incrementally* **Lina Vinikienė, Valentina Dagienė**. *Different Cultures – Different Approaches to Reasoning and Algorithms*
11^{00}–11^{30}	Coffee break	

11^{30}–13^{30}	Session chair: **Jacqueline Staub** **Paper session 6:** room 301 **Programming education** **Jiří Vaníček.** *Concept-building Oriented Programming Education* **Ungyeol Jung, Young Jun Lee.** *The Direction and Possibility for Social Justice in Informatics Education based on Bebras Challenge in Republic of Korea* **Ken Kahn, Niall Winters.** *AI Programming by Children* **Elisabeth Wetzinger, Gerald Futschek, Bernhard Standl.** *A Creative Learning Sequence in an Introductory Programming MOOC*	Session chair: **Wolfgang Slany** **Paper session 7:** room 302 **Robotics** **Julian M. Angel-Fernandez, Nikoleta Yiannoutsou, Chronis Kynigos, Carina Girvan, Markus Vincze.** *Towards a Framework for Educational Robotics* **Flavio Campos.** *Design Curriculum for Educational Robotics: Constructionist Pedagogical Experience in Formal Education* **Dave Catlin, Martin Kandlhofer, Stephanie Holmquist, Andrew Paul Csizmadia, Julian M. Angel-Fernandez, John-John Cabibihan.** *EduRobot Taxonomy and Papert's Paradigm* **Karolína Mayerová, Michaela Veselovská.** *How Students Struggled with Preparation of Activities for a Leisure Time Robotic Workshop*	Session chair: **Ana Isabel Sacristán** **Paper session 8:** room 306 **Jose Armando Valente, Paulo Blikstein.** *The Construction of Knowledge in Maker Education: A Constructivist Perspective* **Panel discussion III:** room 306 *Constructionism across Cultures: Commonalities and Differences of Constructionist Implementations Around the World* joined with papers **Jose Armando Valente, Paulo Blikstein.** *Constructionism in Different Cultures: the case of Brazil &* **Deborah Fields, Paulo Blikstein.** *What Is Constructionism? Views from a Thai Perspective* Jose Armando Valente (moderator), Paulo Blikstein, Deborah Fields, Michael Tan
13^{30}–14^{30}	Lunch		
14^{30}–22^{00}	Excursion & Dinner		

Friday, August 24 / Location: The Faculty of Philosophy (Universiteto St. 9)

8^{30}–10^{30}	Session chair: **Ivan Kalaš** **Plenary session VII:** room 301 **Paul Goldenberg.** *Teaching Children to be Problem Posers and Puzzle-creators in Mathematics* **Ana Isabel Sacristán.** *Constructionist Experiences for Mathematics across Educational Levels*	Session chair: **Jose Armando Valente** **Plenary session VIII:** room 302 **Wolfgang Slany.** *Rock Bottom, the World, the Sky: Catrobat, an Extremely Large-scaling and Long-term Visual Coding Project Relying Purely on Smartphones* **Gary Stager**, co-speaker **Sylvia Martinez.** *Turning Theory Into Practice – Spreading Constructionism*
10^{30}–11^{00}	**Paper session 9:** room 301 **Tiina Partanen, Pia Niemelä, Timo Poranen.** *Racket Programming Material for Finnish Elementary Math Education*	**Demo session 1:** room 302 **Markus Klein, Clemens Koza, Wilfried Lepuschitz, Gottfried Koppensteiner.** *Hedgehog: A Versatile Controller for Educational Robotics*
11^{00}–11^{30}	Coffee break	

11^{30}–13^{30}	Session chair: **Einari Kurvinen** **Paper session 10:** room 301 **Designing activities** **Yasmin Kafai, Deborah Fields.** *Some Reflections on Designing Constructionist Activities for Classrooms* **Kit Martin, Michael Horn, Uri Wilensky.** *Ant Adaptation: A Complex Interactive Multitouch Game About Ants Designed for Museums* **Michael Weigend, Fenja Göcking, Alexander Knuth, Patrick Pais Pereira, Laura Schmidt.** *Media Parkour – Experiential Learning Activities for Media Education* **Brendan Tangney, Ian Boran, Tony Knox, Aibhín Bray.** *Constructionist STEM Activities Using the Bridge21 Model*	Session chair: **Mikko-Jussi Laalso** **Paper session 11:** room 302 **Teacher education** **Daniel Hickmott, Elena Prieto-Rodriguez.** *To Assess or Not to Assess: Tensions Negotiated in Six Years of Teaching Teachers about Computational Thinking* **Daniel Hickmott, Elena Prieto-Rodriguez.** *Constructionist Experiences in Teacher Professional Development: A Tale of Five Years* **Igor Verner, Khayriah Massarwe, Daoud Bshouty.** *Ethnomathematics in Teacher Education: Analysis and Construction of Geometric Ornaments* **Xiaoxue Du, Kay Chioma Igwe.** *Computational Thinking in Teacher Professional Development Programs*	Session chair: **Don Passey** **Paper session 12:** room 306 **Methodologies** **Sven Jatzlau, Ralf Romeike.** *How High is the Ceiling? Applying Core Concepts of Block-based Languages to Extend Programming Environments* **Anton Chukhnov, Sergei Pozdniakov, Ilya Posov, Athit Maytarattanakhon.** *Analysis of Constructive and Cognitive Activities of Participants in Online Competitions in Computer Science* **Vladimiras Dolgopolovas, Valentina Dagienė, Eglė Jasutė, Tatjana Jevsikova.** *Design Science Research for Computational Thinking in Constructionist Education: A Pragmatistic Perspective* **Aleksandra Klašnja-Milićević, Mirjana Ivanović.** *Learning Analytics in Education: Objectives, Application Possibilities and Challenges*
13^{30}–14^{30}	Lunch		

14^{30}–16^{00}	Session chair: **Mihaela Sabin** **Paper session 13:** room 301 **Curriculum matters** **Eva Klimeková.** *Curriculum Intervention for Learning Programming in Python with Turtle Geometry* **Carol Angulo, Alberto J. Cañas, Ana Gabriela Castro, Leda Muñoz, Natalia Zamora.** *Think, Create and Program: Evolving to a K-9 Nationwide Computational Thinking Curriculum in Costa Rica* **Michael Weigend.** *Coding to Learn - Informatics in Science Education*	Session chair: **Carina Girvan** **Paper session 14:** room 302 **Girls in computing** **Bernadette Spieler, Wolfgang Slany.** *Female Teenagers and Coding: Create Gender Sensitive and Creative Learning Environments* **Caitlin Davey, Sawaros Thanapornsangsuth, Nathan Holbert.** *Making Together: Cultivating Community of Practice in an All-Girl Constructionist Learning Environment* **Sawaros Thanapornsangsuth, Nathan Holbert.** *Exploring Girls' Values and Perspectives in Making for Others*	Session chair: **Wilfried Baumann** **Workshop 3:** room 111 **Corey Brady, Walter Stroup, Tony Petrosino, Uri Wilensky.** *Group-based Simulation and Modelling: Technology Supports for Social Constructionism*
16^{00}–16^{30}	Coffee break		
16^{30}– 18^{30}	Session chair: **Witold Kranas** **Paper session 15:** room 301 **Constructionist environments** **Christos Chytas, Ira Diethelm.** *Designing Constructionist Learning Environments with Computational Design and Digital Fabrication* **Kazunari Ito.** *Pictogramming: Learning Environment Using Human Pictograms Based on Constructionism* **Kazunari Ito, Aoi Yoshida, Takashi Yoneda, Yuichi Oie.** *Human Pictogram Unplugged: Unified Learning Environment of Computer Science Unplugged Using Human Pictograms* **Nobuko Kishi, Mari Yoshida, Minori Yoshizawa, Aoi Yoshida.** *VISURATCTH: Visualization Tool for Finding Characteristics of Teaching and Learning Process of Scratch Programmers*	Session chair: **Michael Weigend** **Paper session 16:** room 302 **Modeling** **Natasa Grgurina, Erik Barendsen, Cor Suhre, Klaas van Veen, Bert Zwaneveld.** *Assessment of Modeling Projects in Informatics Class* **Kit Martin, Gabriella Anton.** *Modeling Time* **Ümit Aslan, Uri Wilensky.** *Agent-based Construction (a-b-c) Interviews: A Generative Case Study* **Yu Guo, Uri Wilensky.** *Mind the Gap: Teaching High School Students about Wealth Inequality through Agent-Based Participatory Simulations*	Session chair: **Brian Harvey** **Workshop 4:** room 111 **Ken Kahn.** *AI Programming in Snap!* **Workshop 5:** room 111 **Stephen Howell, Neeltje Berger, Peter Heldens, Kevin Marshall, Clare Riley.** *Developing Affordable STEM Maker Projects with BBC Micro:bits and Microsoft MakeCode*

Saturday, August 25 / Location: The Faculty of Philosophy (Universiteto St. 9)

8^{30}–10^{30}	Session chair: **Deborah Fields** **Plenary session IX:** room 301 **Chronis Kynigos.** *In Support of Integrated Approaches to Constructionist Designs and Interventions: The Case of ChoiCo and MaLT* **Arthur Hjorth.** *Social Gears - a Constructionist Approach to Social Studies*	Session chair: **Paul Goldenberg** **Plenary session X:** room 302 **Brian Harvey.** *May I Teach an Algorithm?* **Jens Mönig.** *Bones, Gears and Witchcraft*

10^{35}– 11^{30}	Session chair: **Eugenijus Kurilovas** **Demo session 2:** room 301 **Nevin Akcay, Hulya Avci, Ali Güngör, Tufan Adiguzel.** *The Relationship between Computer Programming and English Language Skills* **Monica Chan, Gary Lee.** *Synthesizing the Mesh: Using Constructible Authentic Representations to Gain Intuitive Understanding of Bayesian Reasoning*	Session chair: **Christos Chytas** **Demo session 3:** room 302 **Ken Kahn.** *Interpolating (and Extrapolating) 3D Turtle Programs in Beetle Blocks* **Stephen Howell.** *Teaching Computational Thinking with Minecraft & Microsoft MakeCode*	Session chair: **Ralf Romeike** **Workshop 6:** room 111 **Brian Broll, Corey Brady, Ákos Lédeczi.** *NetsBlox: A Constructionist Environment for Creating Distributed Applications*
11^{30}–12^{00}	Closing: Farewell buffet / Location: Grand Courtyard		
13^{00}–20^{00}	Post Conference Excursions (not included in conference fee)		

CONTENTS

OPENING / PLENARIES

One Hundred Years of Educational Development in Lithuania

Rimantas Želvys *rimantas.zelvys@fsf.vu.lt*
Vilnius University, Lithuania

Abstract
This year we celebrate the 100th anniversary of declaring the independent Republic of Lithuania, and the main focus of our attention is the development of education since 1918, which can be divided into at least four different periods.

1918-1940. The period of building. After declaration of the independence in February 1918, the Ministry of Education was established. After the devastating World War I the Ministry discovered in the territory of the newly founded state 8 functioning gymnasiums (upper secondary schools) and 11 progymnasiums (lower secondary schools) with 360 teachers. Besides that, 1232 teachers worked in primary schools, so the whole teachings corps in the country was 1592 teachers. There were no institutions of higher education as the only university in the country – Vilnius University – was closed down in 1832. The system of education had to be built practically from nothing. Lithuanian university was re-established in 1922. Compulsory primary education – four years – was introduced during the period of 1928-1931. In two decades of intense work the fully-functioning system of education was created and the illiteracy rate in 1940 dropped down to 2 percent.

1940-1944. The period of destruction. The Soviet occupation in 1940 and the World War II had an enormous destructive effect on our educational system. Just in a single day – June 14, 1941 – 11 percent of all Lithuanian teachers were deported to Siberia. Many of the teachers died during the war, perished in exile or fled to the West in fear of repressions when the Red Army was approaching. In 1943 the Nazi German authorities closed down Vilnius University as an act of revenge for non-cooperating with the occupational administration. The system of education met the end of the World War II with demolished schools and few remaining teachers.

1944-1990. The period of adaptation. After the war the country had to adapt to the imposed Soviet model of education. Mass education was one of the essential elements of the model. In 1949 compulsory seven-year education, and in 1958 – compulsory eight-year education was introduced. In 1986 the eleven-year long general secondary education was extended to twelve-year long general secondary education. Massification of general secondary schooling was accompanied by the centralization, monopolization, unification, and, most important, strong ideologization of education.

1990-2018. The period of transformation. There are many challenges facing our education. We have to deal with an ideological challenge – what will substitute the previously imposed communist ideology? Do we accept the new "global ideology" – neoliberalism – or shall we look for something else? We face a strategic challenge – what are the long-term goals and mission of our education? Do we accept the prevailing outlook that education is a service which has to supply the global labour market with a necessary workforce, or is it something else? We face a structural challenge – due to demographic reasons is shrinking instead of expansion. We came to the understanding that closing down schools is more difficult than building the new ones. We face the challenge of economical efficiency – how to achieve the desired level of quality and equity in education at the costs which are both available and acceptable to our society? We acknowledge that the current state of our education is far from perfect and that there are still many questions to be answered and many solutions to be found. However, when we look back at the starting position we had one hundred years ago, there is no doubt that we can be really proud with what we have achieved.

CS Unplugged and Computational thinking

Tim Bell, *tim.bell@canterbury.ac.nz*
University of Canterbury, New Zealand

Abstract

The CS Unplugged activities (csunplugged.org) provide a scaffolding for a constructivist approach to introducing topics in computer science, without the need to learn programming first. It has been widely used to support Computational Thinking in school curricula. This paper discusses the connection between CS Unplugged activities and one of the (many!) definitions of Computational Thinking, and discusses how it should be used in this context based on research into using the Unplugged approach effectively in education. In addition to considering the value of an Unplugged approach for teaching students, we will look at broader applications, including supporting teachers who are new to the subject, and using it in integrated learning where computational thinking is exercised as part of other curriculum areas.

The CS Unplugged sorting network activity offers many opportunities for curriculum integration

Keywords

CS Unplugged; Computational Thinking; teacher PLD; integrated learning

Constructionism Won, Now What? The Role of Constructionist Research in the Age of Ubiquitous Computing

Paulo Blikstein, *paulob@stanford.edu*
Stanford University, USA

Abstract
Logo was created almost exactly 50 years ago. It might be the right time to take stock on the accomplishments of the last five decades, and the possible directions for the future. The Constructionist community, having rebelliousness in its DNA, has grown used to say that "the revolution has not come yet." We are still far from realizing Papert's vision, but in the last 20 years there has been impressive change in schools and in the discourse around educational innovation.

The first change is on computing itself. Several high-profile initiatives have brought coding into the mainstream of education, with several cities and countries advocating and implementing programming as a mandatory topic in pre-college education. The scale of these initiatives is impressive, and even though they are still in early stages, they represent a clear recognition that coding has finally been accepted as a school topic at the highest levels of policy making.

Makerspaces and fab labs are a second phenomenon that has reached surprising popularity in schools in just a few years. Thousands of schools already have well-equipped makerspaces, and even though access is not equitable in many of them, their mere presence in schools point to a crucial recognition of the value of constructionist pedagogies, creativity, student agency, and construction.

Third, even traditional disciplines and national standards are being "infected" by the constructionist virus: many science and math curricula around the world now employ constructionist-inspired pedagogies and principles, and some go as far as incorporating tools such as computational modeling and sensing to science classes. And this is also happening at the national level: for example, in the United States, the Next Generation Science Standards made engineering and design mandatory in basic education. And finally, Constructionist tools such as Scratch, NetLogo, Lego Robotics, GoGo Boards, and the Lilypad, have become much more robust and been use by millions of children worldwide.

Given all the good news, what is the right reaction from this community? Claim "mission accomplished" or double down our efforts? It seems that the main challenge for the next 50 years will not anymore convincing schools that many of these technologies and approaches are useful and effective but will be concentrated in two clusters: (a) Making sure that those new learning opportunities are offered to students with equity, and (b) Battling the forces of trivialization, that for economic or ideological reasons, often try to overly simplify the technologies and methods of Constructionism. To overcome these challenges in the next 50 years, we will have to find innovative forms of doing research in learning, new avenues for public advocacy, and novel ways to reach students.

Keywords
Logo; constructionism; computing education; maker education.

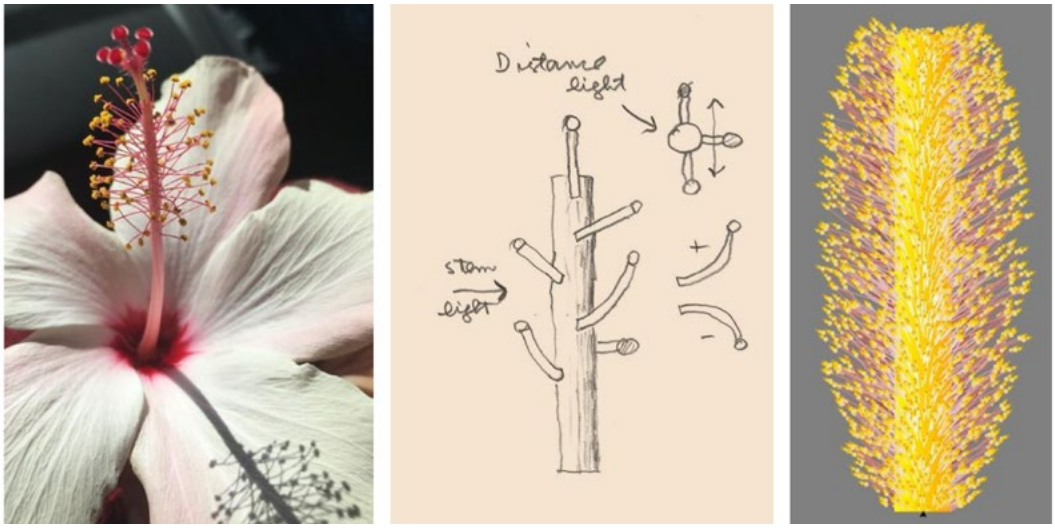
Look Closely, Watch What Happens: Visual Modelling and Constructionism

James Clayson, *james@clayson.org*
American University of Paris, France

Abstract
The core activity in my approach to visual modelling is a series of exercises that encourage students to embark on individual computational explorations of shape, placement, color and texture themes by looking closely at physical objects meaningful to them. In this paper, I will describe the journey that led me to designing and teaching courses around this central notion and how such modelling works, encouraging students to integrate a wide range of technical and non-technical skills into their work. I will talk about why I think this activity is important for students in all disciplines, including the liberal arts. I will talk specifically about the effects visual modelling has had on my students and I will also describe how the teaching and practice of visual modelling have informed my own development and learning over the past 40 years.

My reasons for integrating greater visual imagery into computational activities are summarized in the form of 13 arguments.

Keywords
constructionism, transformational objects, visual modelling, visual thinking, computational tools, liberal arts, Logo, Python



Example: Look, record, see what happens

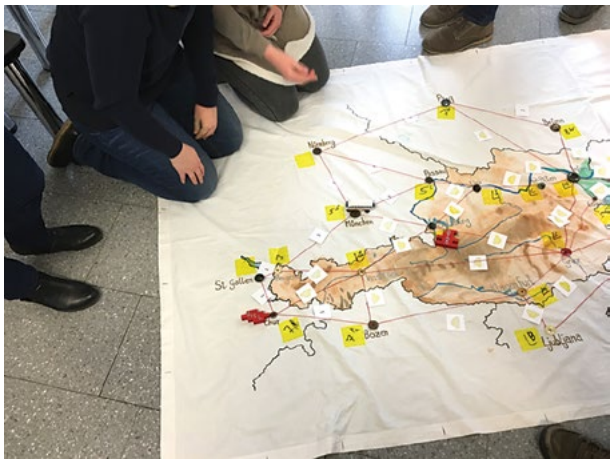
Computational Thinking and Creativity

Gerald Futschek, *gerald.futschek@tuwien.ac.at*
Vienna University of Technology, Austria

Abstract
Creativity is one of the characteristics in constructionist learning. Creativity is frequently connected to learning by doing and needs a high amount of freedom in choice of activities and learning steps. Basically, Computational Thinking denotes thinking processes that are related to problem solving known from computer science. Computational Thinking does not only involve algorithmic thinking skills that are useful in programming and algorithm design but also integrates skills like abstraction, decomposition, generalization and evaluation that are used in problem definition, system modelling and system evaluation.

Since Computational Thinking has become part of informatics school curricula in many countries, the paradigm of competence orientation has led to very detailed curricula that describe a large variety of detailed competences. The high amount of details can lead to a kaleidoscopic teaching practice. Although, from a constructionist viewpoint it seems more promising to learn with learning settings that allow creativity, fun and sense of achievement. The role of creativity in Computational Thinking Learning is not only in creating a creative output but also in finding new ways of thinking to find solutions to problems.

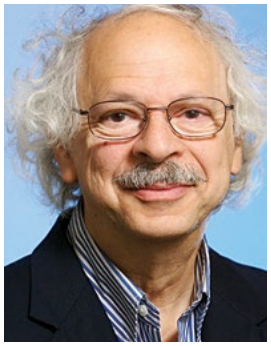
We show several examples of learning settings and didactic projects that allow creativity, fun and sense of achievement in learning Computational Thinking.



Students creating a new game for Computational Thinking Learning

Keywords
Computational Thinking Education, Problem Solving, Creativity

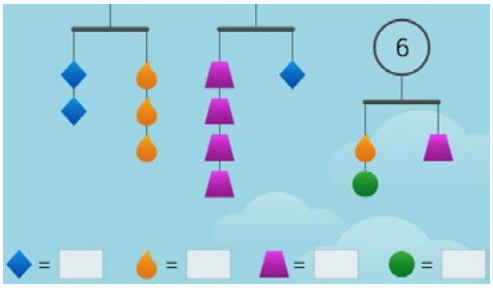
Teaching Children to be Problem Posers and Puzzle Creators in Mathematics[1]



Paul Goldenberg, *pgoldenberg@edc.org*
Education Development Center (EDC), Waltham, MA, USA

Abstract

Seymour Papert's 1972 paper "Teaching Children to be Mathematicians Versus Teaching About Mathematics" started with the summary statement "The important difference between the work of a child in an elementary mathematics class and that of a mathematician is not in the subject matter...but in the fact that the mathematician is creatively engaged...." Along with "creative," a key term Papert kept using is *project* rather than the common notion of *problem*. A project is not simply a very large problem. It centrally includes a focus on sustained and active engagement. The projects in his illustrations were essentially *research* projects, not just multi-step, fully-prescribed, build-a-thing tasks, no matter how nice the end product might be. A *mathematical* playground with enough attractive destinations in it draws children naturally to pose their own tasks and projects—as they universally do in their other personal and group playgrounds—and to learn to act and think like mathematicians. They even acquire conventionally taught content through that play. Physical construction was always available, and appealed to such thinkers as Dewey, but for Papert computer programming, newly available to school, suggested a more flexible medium and a model for an ideal playground.



The instructions you give.	Pictures	Suri
Think of a number.	Imagine your number is hidden in the bag.	
Add 3.		
Double that.		16
Subtract 4.		12
Divide by 2.		
Subtract your original number.		

Two puzzles that introduce algebra's logic and then its notation. Children can invent their own.

A fact about playgrounds is that children *choose* challenge. In working and playing with children I've seen that puzzles tap some of the same personally chosen challenge that a programming-centric playground offers. Children are naturally drawn to intellectual challenges of riddles and puzzles (ones they learn and ones they invent); and adults are so lured by puzzles that even supermarkets sell books of them. So what's the difference between real puzzles and school problems? What's useful about *creating* a puzzle or posing a problem? How might puzzles and problem posing support mathematical learning? And what's constructionist about this? This plenary will try to respond to these questions, invite some of your own responses, let you solve and create some puzzles, and explore how problem posing in programming and puzzling can support mathematics even in an age of rigid content constraints.

Keywords

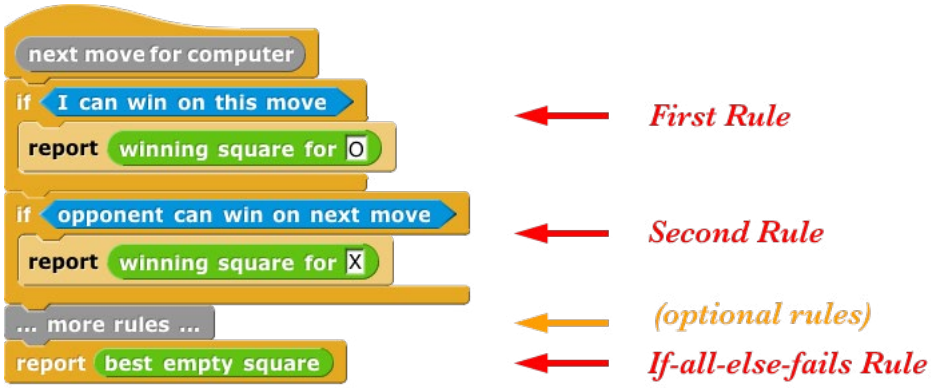
Problem posing; puzzles; mathematics; algebra

[1] Funding for doing and reporting the work described in this paper was provided in part by the National Science Foundation, grants 1441075, 1543136 and 1741792. Views expressed here are those of the author and do not necessarily reflect the views of the Foundation

May I Teach an Algorithm?

Brain Harvey, *bh@berkeley.edu*
University of California, Berkeley, USA

Abstract
After all these years, I'm still not sure what Constructionism entails. Here's an example taken from our work on the *Beauty and Joy of Computing* curriculum: We have a yearlong Tic-Tac-Toe project. Early in the year, students draw the board and use mouse clicks on the board to let two human players alternate moves. A month later, they start building data structures that will let the program analyze the board. At that time, they check whether either player has won, or whether the game is tied. Two months after that, they return to the project, letting the program be one of the players, and determining the program's move with rules such as "if I can win on this move, do it." A Tic-Tac-Toe program is an obvious project, one of the things many learners do spontaneously. But I had an ulterior motive in the design of the project: I want students to practice using higher order functions (MAP, KEEP, COMBINE). The data structures in the program are built to accommodate that.



We provide the overall structure of the strategy procedure.

The question is, because I have this motive and I impose this design, am I being hopelessly instructionist? Is it *my* project rather than the kid's project? How much flexibility is required for the kid to "own" the project? And, is the kid owning the project what makes it Constructionist?

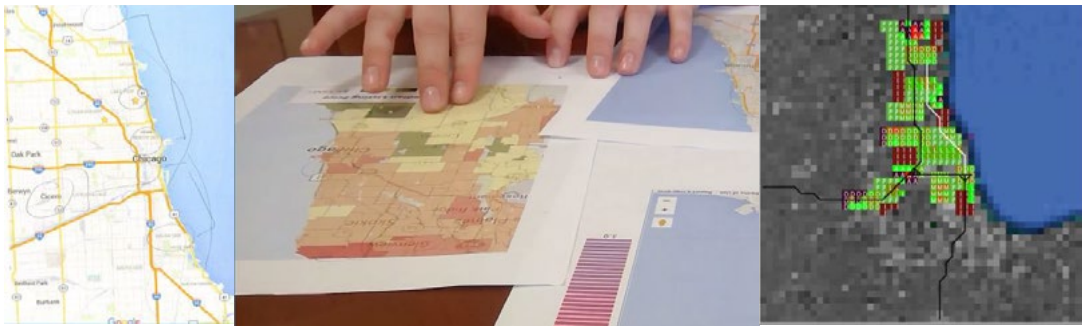
Keywords
curriculum; ownership; constructionism; instructionism

Social Gears – a Constructionist Approach to Social Studies

Arthur Hjorth, *arthur.hjorth@u.northwestern.edu*
The Center for Connected Learning and Computer-Based Modeling, Northwestern University, USA

Abstract
The Gears of Seymour Papert's childhood have been a persistent analogy for our work in the Constructionist community for decades. It is a productive analogy that has helped concretize many central themes in Constructionism: It frames our view of knowledge as understanding the inner workings of a system; it forefronts the existence of an *external representation* of some domain knowledge; and it emphasizes that learning happens through the manipulation of this external representation by alignment of an internal, mental model with an external, physical model. Gears are powerful, maybe *because* their cause-and-effect is simple: one cog moves another cog which moves another cog which moves another cog. Always at the same ratio, and at a rate pre-determined by whichever cog we apply torque to. Importantly to our work as an educational research community, it makes studying *thinking* about cogs relatively straight forward.

But what if the gears were *social*? What if they have inner lives, mood swings, wants and desires, and work under the constraints of social pressures and modern family life that they must negotiate, collectively and individually, in order to organize their turn ratios? Does it change how we should design Constructionist learning environments and activities? Does it change how we should study students' thinking? I will not claim that the deterministic nature of the Gears-analogy *caused* Constructionism to focus on deterministic subjects. But maybe the focus on STEM and programming have led us to not explore and interrogate potential shortcomings of the analogy?



Students reasoning with different representations of Urban Planning

I address these questions based on my experience with designing, implementing, and studying Constructionist learning in social studies classrooms at the high school level. I present "Complex *Social* Systems Thinking" (CSST) as a guiding framework for designing learning activities and for studying students' thinking. I then present a set of Constructionist learning activities that I have designed on Urban Planning, and present data to exemplify what CSST looks like "in the wild". Finally, I discuss the relationship between Constructionism and CSST and present my hopes for the future of Constructionism in social sciences education.

Keywords
complex systems thinking; social studies; design; qualitative analysis

Scratchmaths: a Positive Outcome for Constructionism at Scale

Richard Noss, *r.noss@ucl.ac.uk*
University College London, UK

Celia Hoyles, *c.hoyles@ucl.ac.uk*
University College London, UK

Abstract

From September 2014, all primary schools in England have been required to teach the national computing curriculum, which includes designing and building programs. This presentation will discuss some of the challenges in implementation as part of describing the two-year ScratchMaths (SM) project. SM designed a comprehensive curriculum for Year 5 and 6 pupils (aged 9-11 years) that maps directly to the computing curriculum, seeks to develop pupils' programming skills as well as exploit these skills to explore key mathematical concepts. The SM curriculum comprises detailed student activities for about 20 hours per year over two years, with teacher support materials and professional development days. SM has been implemented in over 100 schools across England. The project was evaluated qualitatively through design, survey and observational research. In addition, SM was evaluated by an independent research team employing a randomised control trial.

We will report a surprising result on the constructionist path from the RCT, namely that:

> **ScratchMaths led to a small increase in computational thinking as measured by the test used at the end of the first year of the trial. (draft independent report on SM project).**

This is surely good news for the constructionist community, not least as it was shown that this increase was significantly greater among 'disadvantaged students' as measured by their eligibility for free school meals (a standard proxy measure in UK). We will explore the substantive issues concerning this finding and the quantitative methods employed such as: What can a Scratch-aware learner do that he/she couldn't have done without Scratch?'. And what is meant by computational thinking.

We will also report the second major finding of the SM project, namely that:

> **...even though a positive correlation was found between computational thinking and mathematics, ScratchMaths did not increase mathematics attainment during the trial period, as measured by Key Stage 2 tests. (draft independent report on SM project)**

Again, there is much to say to seek to explain this outcome: it raises questions again of the validity of the measures, but also of teacher confidence and the 'fidelity' of the intervention as implemented in the second year. Educational interventions are complex and it is difficult to be clear what are the key components of a given intervention, that transcend any particular context. These outcomes give us all we think food for thought and we note we would have missed them if we had closed the door on methodologies that have, up to now, been relatively taboo in constructionist circles.

Keywords

Scratch; computational thinking; mathematics; methodology; randomised control trial

References

Benton, L. Hoyles, C., Kalas, I & Noss, R. (2017) Bridging Primary Programming and Mathematics: preliminary findings of design research in England Digital Experiences in Mathematics Education, pp 1- 24

Benton, L. Kalas, I; Saunders, P; Hoyles, C; Noss, R. (in press) Beyond Jam Sandwiches and Cups of Tea: An Exploration of Primary Pupils' Algorithm-Evaluation Strategies" J of Computer Assisted Learning

Programming in Lower Primary Years: Design Principles and Powerful Ideas

Ivan Kalaš, *kalas@fmph.uniba.sk*
Comenius University, Bratislava, Slovakia

Abstract
Latest national computing education strategies – often installing the beginning of the new mandatory subject into the lowest years of primary education – embrace programming as a key instrument of computational thinking. National curricula usually set ambitious requirements for primary computing education, listing basic *computational constructs* and *practices* to be mastered by primary pupils (e.g. *use sequence, selection, and repetition in programs;* or *design, write and debug programs that accomplish specific goals...*). While the research findings within our recent ScratchMaths project suggest that the intervention which we developed is a viable strategy to meet the expectations of national curricula in years 5 and 6, the question remains how to implement them in years 1 to 4 (i.e. with the age group of around 5 to 10, depending on educational system).

There are numerous portals and on-line resources claiming to have the answer to that question. Our main concern, however, is that those resources and the expertise behind them often originate from after-school experience, secondary or higher education practise or individualised home "edutainment" – focusing on isolated flashes of learning, often neglecting complexity of important basic computational constructs and practices, plus failing of advantages of primary education.

In our on-going research and development, we strive to better understand what distinguishes after-school programming environments and approaches (in the *code.org* style) from systematic and appropriate pedagogies for lower primary computing. In my plenary I will present our emerging approach for transforming so called 'basic' *computational constructs* into thoroughly constructed and iteratively verified gradation of short units of programming tasks which the pairs of pupils – and then the entire class – try to explore and solve, envisage and discuss, compare, share and explain, exploiting the 5Es pedagogical framework of the ScratchMaths project. I will give reasons why we decided to develop new set of programming environments with Emil, see the figure, to implement our strategy and I will formulate our key design principles and explain which powerful ideas we want pupils to experience and explore so that they get the opportunity to gradually build deep understanding of several *computational constructs* (or *pre-constructs*) and *practices* in appropriate progression.

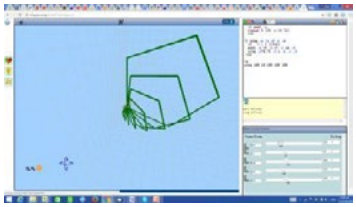
Different styles of control of Emil with increasing cognitive demand

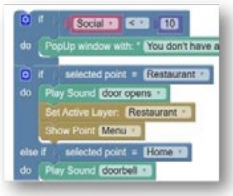
Keywords
Primary Programming; programming environments for primary pupils; developmental appropriateness; Email

In Support of Integrated Approaches to Constructionist Designs and Interventions: The Case of ChoiCo and MaLT

Chronis Kynigos, *Kynigos@ppp.uoa.gr*
National and Kapodistrian University of Athens, Educational Technology Lab School of Philosophy, P.P.P. Dept, Greece

Abstract
Constructionism is now a 50 year-old theory of learning, a theory of educational design and a framework for pedagogical action. In this time, society, educational challenges and the abundance of digital media have brought a diversity of frames, focal points, viewpoints and interventions. In many places silo constructionist perspectives are seen as obsolete or at best ultimate frames for meaning-making through individual and social bricolage. In my talk I will argue for perspectives integrating constructionism in a wider landscape of educational paradigms, theories, affordances and intervention strategies. I will do this by showing what students and teachers have built with two web-based constructionist expressive media, one very different to the other, MaLT – turtle sphere and 'Choices with consequences' games (ChoiCo). In my context these are proving to be powerful means for a proximal approach to infusing constructionist perspectives in wide scale initiatives.





MALT2 - programming, 3D, Dynamic Manipulation, http://etl.ppp.uoa.gr/malt2
ChoiCo: GIS, Programming, BD, http://etl.ppp.uoa.gr/choico

MALT is a Logo programmable 3D simulator including Turtle Graphics and most importantly dynamic manipulation of variable values with DGS - like effects on the graphical output from variable procedures. ChoiCo is a tool for game modding, supporting a socio-scientific paradigm of diverse consequences games involving complex issues like dietary choices or environmental issues and embedding powerful ideas in various ways. The examples will show how avenues for powerful ideas uniquely made available and embedded in multifaceted issues present exciting challenges for design.

Keywords
integrations, affordances, educational paradigms, programming

Bones, Gears and Witchcraft

Jens Mönig, *jens@moenig.org*
Principle Investigator SAP, Germany

Abstract
Who doesn't like a good story? I never get tired playing with stories kids animate with Scratch or Snap! To me, storytelling is at the heart of the current digital literacy movement. But it's not just about kids. I've been surprised by the culture of storytelling nurtured in big industrial companies. Designing the right story can be crucial for a project, a program or a promotion. The opposite is also true: The wrong story has the ability to compromise funding and even thwart a career. And then there are stories that convince for all the wrong reasons, and success that feels like defeat.

I will share a few of my stories for children, corporate management and government officials. Among them, how Katharina Kepler's witchcraft trial has been a turning point for computing, and how machine learning can be used to illustrate a business proposal, before examining a particular terrible instance of constructionism gone wrong in the German state of Baden-Württemberg. Expecting this to spark some controversy I will close by opening up a discussion with the audience about favorite stories, inviting examples of "good" and " bad" specimens..

Keywords
Scratch, Snap!, ldigital literacy

Constructionist Experiences for Mathematics Across Educational Levels

Ana Isabel Sacristán, *asacrist@cinvestav.mx*
Centre for Research and Advanced Studies, Cinvestav, Mexico

Abstract

Constructionism, in the research literature, is seen predominately in contexts where it is used as a paradigm for promoting learning at K-12 levels. However, it can apply at all levels, as well as connect levels. In this paper I deal with the issue of classroom implementations for mathematical learning of constructionist experiences at different educational levels, but with particular focus on higher education implementations. I begin by revisiting some of the fundamental principles of constructionism. Then, I look briefly at how constructionism can provide early access to powerful ideas; that is, where younger learners can access mathematical ideas perceived as more advanced. For that, I mention my work in the design of two different microworlds that enabled young students to explore and engage with mathematical infinity-related ideas.

I then share some of my experiences in several attempts of constructionist implementations and microworlds for mathematical learning at university level. I present two examples that involved building and exploring computer models and simulations of real phenomena: the first in a distance-learning course conducted as a virtual mathematics laboratory; the second, of videogame construction by engineering students (see figure below). A third example is of computer programming R-based tasks for the learning of statistics in environmental science students. I finish by presenting a fourth example from a university in Canada where mathematics university students are required to program digital mathematical exploratory objects or microworlds. In all the examples presented, students engage in constructing models or programs, and in doing so, engage in doing mathematics.

	Player	Gift Box	Motorcycle	Jeep	Gem
Density	0.2	0.01	0.5	0.7	0.8
Restitution	0.8	0.01	0.1	0.1	0.01
Linear damping	0.1	0.3	0.4	0.5	0.9
Angular damping	0.01	0.1	0.5	0.7	0.6
Friction	0.2	0.01	0.5	0.7	0.9

```
var fixture = physics_fixture_create();
physics_fixture_set_circle_shape(fixture, 16);
physics_fixture_set_density(fixture, 0.2);
physics_fixture_set_friction(fixture, 0.2);
physics_fixture_set_restitution(fixture, 0.8);
physics_fixture_set_linear_damping(fixture, 0.1);
physics_fixture_set_angular_damping(fixture, 0.01);
physics_fixture_set_collision_group(fixture,1);
physics_fixture_bind(fixture, inst_D4F006DC);
physics_fixture_delete(fixture);
```

A students' videogame (left), with his table of physical characteristics of the objects involved (top-right) that he would need to program into the game engine (bottom-right).

Keywords

Constructionism; computer programming and expressive media; collaborative learning; modelling; higher education.

Back 100 000 (2)

Evgenia Sendova, *jenny.sendova@gmail.com*
Institute of Mathematics and Informatics at the
Bulgarian Academy of Sciences, Bulgaria

Abstract
I am sharing memories of the first conference *Children in Information Age*, held in Varna (Bulgaria) 32 years ago – both personal as well as extracted from the proceedings and two documentaries featuring the vision of eminent scientists and educators on computers in education from different continents. Based on discussions with researchers, teachers, parents, representatives of industry, discussed herein are their questions and visions in today's context. Below are some of the questions posed then, still relevant today:

In what way should the students be trained as to the future problems of rational use of computers? How can we make the best use of the computer in the future?

- Shall we use computers to make the educational process more technical or more human?
- What does "computer literacy" mean in fact? How should we define literacy?
- Which statements made today would be obsolete in a few years, if not months?
- Should the computer be seen as an attraction in itself holding a child indoors with its artificial simulations, however realistic they might be?
- What are the new roles of the educators?
- What might we really like to see?

After considering the views of the pioneers of the information age I am sharing some concerns based on a recent experience in observing uses of technology dictated mainly by the market models in the educational institutions.

The views of representatives of various stakeholders of education are presented as shared in a recent panel discussion on the future education we would like to see happen.

After considering some finding and recommendation published in a document of the European Parliament on the teaching and learning in the Digital era I conclude with an optimistic perspective based on the work with teachers and students in the frames of national and European educational projects in which I have been involved.

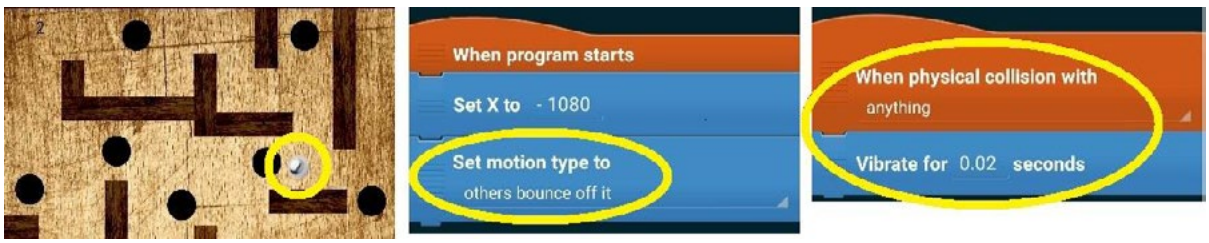
Keywords
Teaching and learning in the information age, digital era

Rock Bottom, the World, the Sky: Catrobat, an Extremely Large-scale and Long-term Visual Coding Project Relying Purely on Smartphones

Wolfgang Slany, *wolfgang.slany@tugraz.at*
Graz University of Technology, Austria

Abstract

Most of the 700 million teenagers everywhere in the world already have their own smartphones, but comparatively few of them have access to PCs, laptops, OLPCs, Chromebooks, or tablets. The free open source non-profit project Catrobat allows users to create and publish their own apps using only their smartphones. Initiated in 2010, with first public versions of our free apps since 2014 and 47 releases of the main coding app as of July 2018, Catrobat currently has more than 700,000 users from 180 countries, is available in 50+ languages, and has been developed so far by almost 1,000 volunteers from around the world ("the world"). Catrobat is strongly inspired by Scratch and indeed allows to import most Scratch projects, thus giving access to more than 37 million projects on our users' phones as of July 2018. Our apps are very intuitive ("rock bottom"), have many accessibility settings, e.g., for kids with visual or cognitive impairments, and there are tons of constructionist tutorials and courses in many languages. We also have created a plethora of extensions, e.g., for various educational robots, including Lego Mindstorms and flying Parrot quadcopters ("the sky"), as well as for controlling arbitrary external devices through Arduino or Raspberry Pi boards, going up to the stratosphere and even beyond to interplanetary space ("the sky"). A TurtleStitch extension allowing to code one's own embroidery patterns for clothes is currently being developed. Catrobat among others intensely focuses on including female teenagers. While a dedicated version for schools is being developed, our apps are meant to be primarily used outside of class rooms, anywhere and in particular outdoors ("rock bottom", "the world"). Catrobat is discovered by our users through various app stores such as Google Play and via social media channels such as YouTube as well as via our presence on Code.org. Sharing, remixing, and collaboration is actively encouraged and supported. Catrobat has a very long term perspective in that it is independent of continuous funding and actively developed in a test-driven way by hundreds of pro-bono volunteers from around the world. Our aim is to grow by a factor of thousand and reach a billion users by 2030. We warmly welcome new contributors in every imaginable field and way with open arms. Please join us and contact me via *wolfgang@catrobat.org* today!



Partial screenshots from "Tilt maze 1.0" that relies on Pocket Code's physics engine, where the physical behaviour of objects is set through its "motion type", e.g., "others bounce off it" for the wooden walls of the maze (middle script), and a short vibration when the ball touches a wall (script on the right). The movement of the ball by tilting the phone is realized by a "Set gravity for all objects to X: -3 x inclination_x Y: -3 x inclination_y steps/second²" brick that is executed in a "Forever" loop (not shown here).

Keywords

pocket code, game design, gaming, gender Inclusion, coding, mobile Learning, social Inclusion, constructionism, girls, teenagers, apps, smartphones, tinkering

Making Constructionism Great Again

Gary S. Stager, *gary@stager.org*
Constructing Modern Knowledge, USA

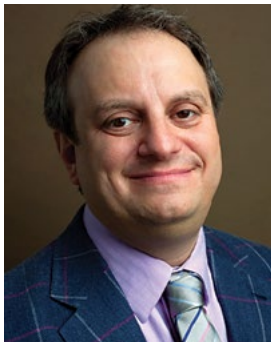
Abstract
The Constructionism community is at a crossroads with the passing of Seymour Papert, the uncertain future of Logo, and the emergence of simpatico movements outside of the academy. This session will explore unfinished work in addition to the challenges and opportunities faced by the next generation of constructionists. What will it take to sustain its relevance and make constructionism great again?

Topics explored include:

- The Premature Death of Logo
- Seeing the "Entire Elephant"
- Piaget versus Popular Coding Curricula
- Twenty Things to Do with a Computer Today
- Agency and the Lost Art of Teaching
- Know Who Your Friends Are
- The Progressive Imperative

Keywords
Progressive education, constructionism, Logo, Seymour Papert, Scratch, physical computing, CS4All

Turning Theory into Practice – Spreading Constructionism



Gary S. Stager, *gary@stager.org*
Constructing Modern Knowledge, USA

Sylvia Martinez, *Sylvia@inventtolearn.com*
Constructing Modern Knowledge, USA

Abstract

In age marked by ascendant instructionism, the presenters have led two initiatives that introduced constructionism successfully to preschool – high school educators around the world. Without compromising the powerful ideas of Papert or his learning theory, the presenters have made constructionism accessible and resonant among practicing educators through publishing and professional development efforts. Countless educators have been inspired bring constructionism to life, often accompanied by the use of cutting-edge technology, without government, foundation, or academic support.

Five years ago, the presenters published *Invent To Learn: Making, Tinkering, and Engineering in the Classroom.* This book is quite possibly the most popular text ever written about constructionism and has been translated into multiple languages. *Invent To Learn* sought to situate the emerging maker movement in a theoretical context of constructionism and historical context of progressive education while building a bridge between the informal learning movement outside of schools and sound classroom practice. Constructionism pervades the text explicitly and tacitly. The success of *Invent To* Learn led to the publication of ten other books by constructionist educators. During this plenary session, the authors will reflect upon lessons learned about learning-by-making, teaching, and school change since the time of publication.

Nearly twenty years ago, Gary Stager and Seymour Papert engaged in multiple conversations about building a different bridge; one between our progressive education colleagues suspicious of modernity and an educational technology community that, in Papertian terms was increasingly "idea averse." While Papert was never able to convene such a summit, Gary Stager created the annual Constructing Modern Knowledge summer educator institute. Over eleven years, Constructing Modern Knowledge has created an immersive learning environment modeling constructionism and pedagogical strategies developed collaborative with Seymour Papert during their "prison project." Papert's unique emphasis on the competence of educators, the absence of coercion, computer as material, powerful ideas, technology as prosthetic, project-based learning, and the centrality of the learner – especially when the learner is a teacher - create the conditions for countless educators to not only develop exceptional computational fluency, but construct personal lessons for creating productive contexts for learning in their personal school contexts.

At the Constructionism 2018 Conference, we will share the unique structure of Constructing Modern Knowledge, along with learning stories and project vignettes supporting the efficacy of constructionism by and for educators willing to take off their teacher hats and put on their learner hats.

Keywords

Constructionism, Logo, Seymour Papert, maker movement, coding, fabrication, professional development, progressive education

The Evolution of a Constructionist Teacher (with Some Reminders from Seymour)

Carol Sperry Suziedelis
Millersville University, USA

Abstract
Many teachers who are aligned with the organic, creative, and dynamic ideas of Constructionism find it difficult to navigate in traditional waters. The obstacles are many: rigid curriculum, excessive testing, lack of resources, few allies, inability to articulate philosophy to the satisfaction of the powers that be. It takes courage, and this paper attempts, through anecdote and narrative, to offer ways and means to develop a Constructionist mindset and the tools and attitudes to effect changes. We hope to inspire discussion of topics such as what it means to learn, thinking about thinking, the importance of teacher engagement, relationships, relevancy, aesthetics, gender issues, and project-based learning.

Along the way, we will remember Seymour Papert, his dedication to the possibilities of "learning as a dimension of life," and resurrect some of the ideas" he used to inspire us.

Keyword
Constructionist teacher, dynamic

Reempowering Powerful Ideas

Uri Wilensky, *uri@northwestern.edu*
Northwestern University, Chicago, IL, USA

Abstract

In Mindstorms, Papert expressed many ideas foundational to Constructionism. Among these are how children programming can be transformative for learning, the empowerment children gain by creative expression with the computer, and powerful ideas of computation. In the nearly four decades since Mindstorms, there has been great progress in realizing these potentials. In particular, there has been wide recognition of the importance of coding and of computational thinking for all and many initiatives have arisen to galvanize these efforts. These new efforts are exciting for constructionists and we have seen public and academic vindication for these ideas.

In this paper, I point out that, in addition to these ideas, there is a part of Papert's vision in Mindstorms, that is more neglected. That is the more general notion of powerful ideas, not solely ideas inherent in computation such as variables, procedures and recursion, but also powerful ways of thinking from other disciplines that are made more powerful and accessible through computation. Mindstorms expounded on turtle geometry, making it accessible both through programming the turtle and engaging with turtle microworlds. In either mode, the turtle added to the Euclidean point one other property, that of a heading, and therefore angular velocity. This change results in a new definition of a circle, one that is intrinsic and so connects geometry to the powerful ideas of calculus. This change in the Euclidean point is a case of what Wilensky and Papert called a restructuration, a change in the representational infrastructure used to encode knowledge. Restructurations throughout history have increased the power, usability and learnability of formerly difficult knowledge. Classic examples are the restructurations of arithmetic from Roman to Hindu-Arabic representation and the restructuration of kinematics from natural language to algebra. Both of these dramatically democratized access to these powerful ideas. Wilensky & Papert argued that, like in turtle geometry, computational representations can serve as the basis for significant restructurations -- restructurations that increase the power and learnability of powerful ideas in science. Like the turtle, a powerful means of creating restructurations is to add agency to primitive elements. This can be achieved through agent-based computational approaches. In this paper, I'm going to show examples of the many different restructurations that members of the CCL lab have constructed in the past decade including NetLogo-based restructurations of powerful ideas of biology, materials science and economics -- and invite the constructionist community to increase its efforts in creating computational representations of powerful ideas. These ideas are accessed through multi-turtle programming and emergent systems microworlds. Agent-based representations provide "objects-to-think with" that facilitate powerful ideas of discrete mathematics, probability and network theory. As the world increases in complexity, citizens of a society increasingly require use of these powerful ideas to make sense of their natural and social worlds and to be empowered to make meaningful changes in society.

Agent-based models of a) artificial selection of sunflowers b) predator-prey ecosystem

Keywords

powerful ideas; restructurations; agent-based modeling, NetLogo

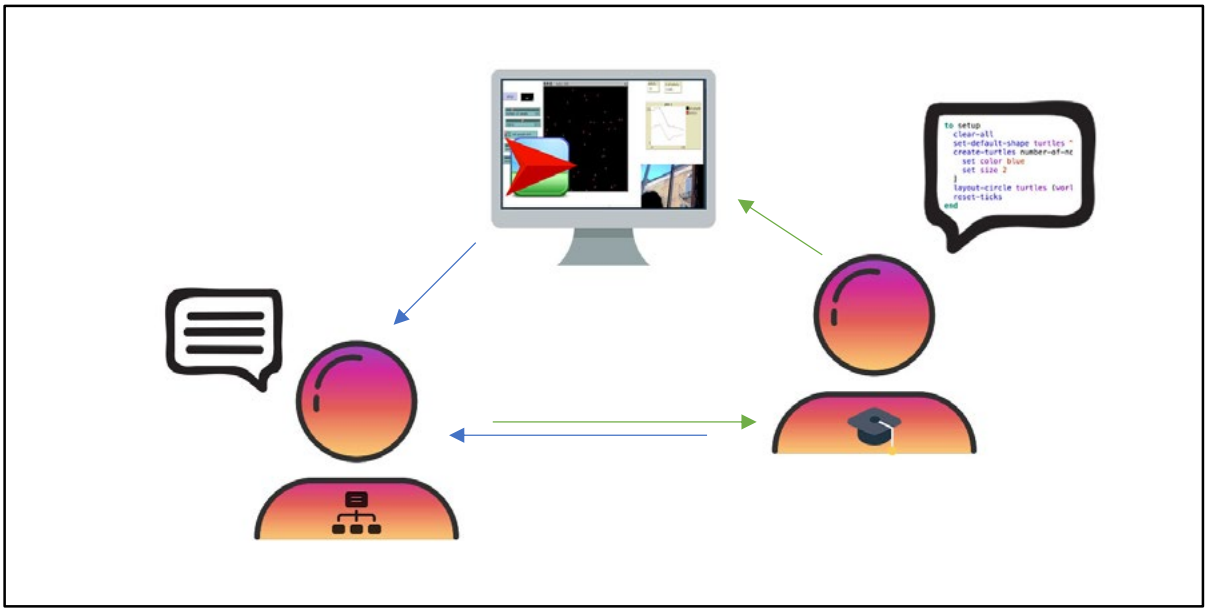
RESEARCH PAPERS

Agent-based Construction (a-b-c) Interviews: A Generative Case Study

Umit Aslan, *umitaslan@u.northwestern.edu*
Learning Sciences, Northwestern University, USA

Uri Wilensky, *uri@northwestern.edu*
Learning Sciences, Computer Science and Complex Systems, Northwestern University, USA

Abstract
We propose agent-based construction (a-b-c) interviews as a new research methodology specifically designed to expose patterns of reasoning about emergent phenomena and complex systems. In an a-b-c interview, the researcher acts as an active mediator between the participant and an agent-based modeling environment. As the participant describes the model, the researcher tries to write the corresponding code and probes the participant about his or her reasoning. In this paper, we present a generative case study in which an adult participant constructs a NetLogo model of aging with the help of a researcher. We conduct a preliminary grounded analysis of this case study and trace the evolution of the participant's model throughout the hour-long interview. Our findings show that the act of mediation between the participant and the agent-based modeling environment can potentially afford, at times even obligate, the researcher to continuously make on-the-fly hypotheses about the participant's thinking, present these hypotheses through writing the model's code, and get immediate feedback from the participant. Our findings also show that a-b-c interviews can potentially expose more fine-grained, spontaneous, and connected reasoning processes that cannot easily be studied through traditional task-based or verbal clinical interviews.



A diagram of an a-b-c interview.

Keywords
agent-based modelling, complex systems, emergence, knowledge, reasoning, research methodologies, clinical interviews

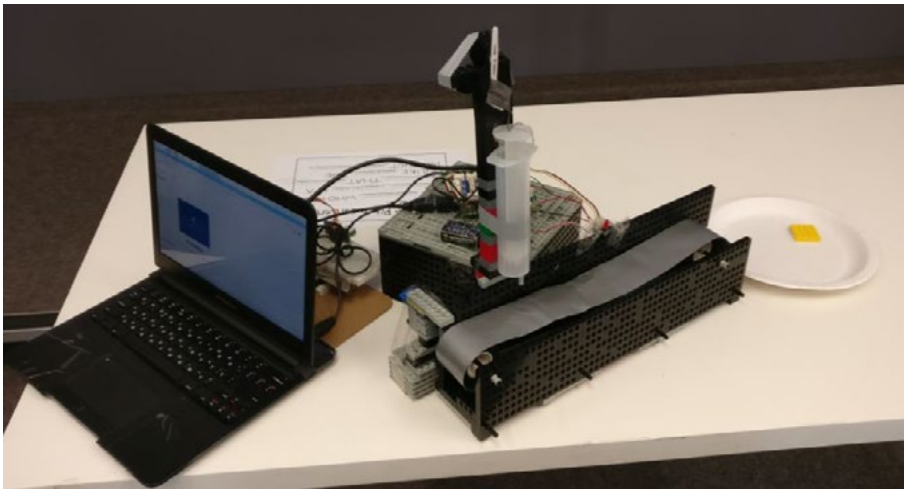
Active Learning of Computer Science Using a Hackathon-like Pedagogical Model

Jake Rowan Byrne, *jake.byrne@tcd.ie*
The Trinity Centre Research for IT in Education, School of Education and School of Computer Science & Statistics, Trinity College Dublin, the University of Dublin, Ireland

Kevin Sullivan, *kevin@bridge21.ie*
The Trinity Centre Research for IT in Education, School of Education and School of Computer Science & Statistics, Trinity College Dublin, the University of Dublin, Ireland

Katriona O'Sullivan, *Katriona.OSullivan@mu.ie*
Department of Adult and Community Education, Maynooth University, Kildare, Ireland

Abstract
Despite extensive literature on Computer Science (CS) education, there are few pragmatic pedagogical models that support active and project based learning. The new Irish CS course for post-primary schools advocates such an active learning approach through a series of "Applied Learning Tasks" (ALT). This paper explores the use of a constructivist/constructionist 21st Century pedagogical model, delivered as a weeklong "Hackathon-like" activity, targeting the learning outcomes for the embedded systems ALT from this new CS course. Twenty-one students participated in the workshop and completed pre and post surveys to assess their confidence in CS topics and their associated learning outcomes. Analysis revealed that students were more confident in almost all the learning outcomes surveyed. There were particularly significant increases in the embedded systems learning outcomes, which was the focus of the students' projects. The findings suggest that the combination of a "Hackathon-like" event and a constructivist/constructionist learning model can be effective in increasing student motivation, confidence and learning of Computer Science concepts and skills at post-primary level.



"Froasties": Participants created an automated food processor, specializing in French toast.

Keywords
Constructivism, constructionism, hackathon, computer science education, creativity, design thinking, problem solving, prototypes, teamwork

EduRobot Taxonomy and Papert's Paradigm

Dave Catlin, *dave@valiant-technology.com*
Valiant Technology Ltd., UK

Martin Kandlhofer, *mkandlho@ist.tugraz.at*
Graz University of Technology. Austria

Stephanie Holmquist, *sh2@usf.edu*
University of South Florida, USA

Andrew Paul Csizmadia, *A.P.Csizmadia@staff.newman.ac.uk*
Newman College, UK

Julian M. Angel-Fernandez, *Angel-Fernandez@acin.tuwien.ac.at*
Technische Universität Wien, Austria

John-John Cabibihan, *john.cabibihan@qu.edu.qa*
Qatar University, Qatar

Abstract
Seymour Papert was the first person to suggest using robots in education. And for nearly 50 years his constructionism principles guided their application in schools. Over the last few years, a multitude of new robots have become available. Do they all meet Papert's Paradigm? What of those that don't? In this paper, we further develop the EduRobot Taxonomy which puts an ever-increasing number of educational robots into order. We'll examine what they've got in common and how they differ: we'll also see if they all comply with Papert's ideas and what does it mean if they don't.

Keywords
Education Robots, EduRobot Taxonomy, Build Bots, Use Bots, Social Robots

Analysis of Constructive and Cognitive Activities of Participants in Online Competitions in Computer Science

Anton Chukhnov, *septembreange@gmail.com*
Saint Petersburg Electrotechnical University "LETI", Russia

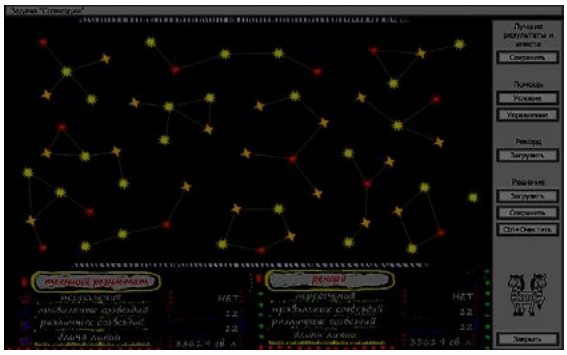
Sergei Pozdniakov, *pozdnkov@gmail.com*
Saint Petersburg Electrotechnical University "LETI", Russia

Ilya Posov, *iposov@gmail.com*
Saint Petersburg Electrotechnical University "LETI", Saint Petersburg State University, Russia

Athit Maytarattanakhon, *seaay2499@gmail.com*
Saint Petersburg Electrotechnical University "LETI", Russia

Abstract

The paper discusses a certain type of competitions based on distance interaction of a participant with simulation models of concepts from discrete mathematics and computer science. The "Construct, Test, Explore" competition, developed by the authors, is chosen to be a representative of such competitions. One of the features of this competition is that all its tasks are accompanied not only with simulation models and tools to manipulate the model's objects, but also with a hierarchical criteria system, that defines an objective function to be optimized while solving a task. The presence of such a criteria system allows for treating a task's subject as a set of several tasks of different complexity (2–4 tasks). Each criterion has means to asses partial solutions, that provide permanent feedback for a participant. The transition of a participant from optimizing one criterion to optimizing the next one, means that he or she has already fully understood an idea corresponding to a former criterion.



On the example of a subject based on the graph theory, the methodology of building subject tasks is demonstrated. The methodology uses a suitable interpretation (metaphor), and means of qualitative and quantitative analysis of solutions, that allow for assessing the formation and understanding by a participant of the graph isomorphism concept, and also several other graph theory concepts. The possibility of using the results of the experiment for assessing the results of cognitive activity, as well as the role of the hierarchical system of criteria for managing cognitive activity, is shown.

The figure below demonstrates the user interface of a subject task based on the idea of constellations to describe graph theory concepts. Participants should create as many non-isomorphic constellations as possible, the second criterion is the total number of constellations (even isomorphic), and the last criterion is the total length of constellations segments.

The work is supported by the Russian Foundation for Basic Research (Project No. 18-013-01130)

Keywords

olympiad, computer science, discrete mathematics, electronic manipulator, CS competition, mathematical thinking

Short Tasks – Big Ideas: Constructive Approach for Learning and Teaching of Informatics Concepts in Primary Education

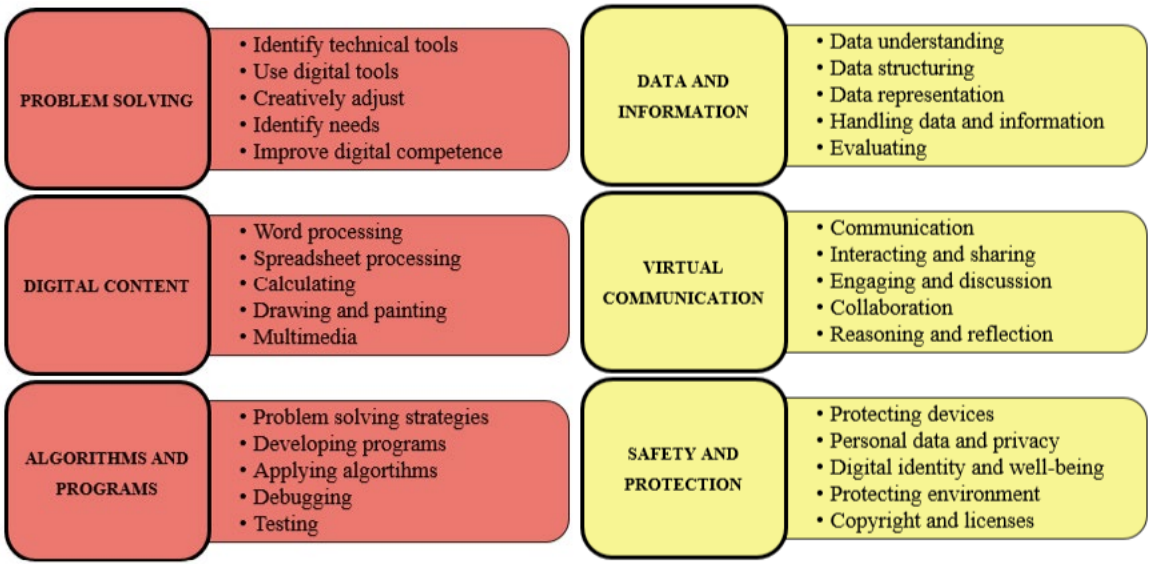
Valentina Dagienė, *valentina.dagiene@mii.vu.lt*
Vilnius University Institute of Data Science and Digital Technologies, Lithuania

Gabrielė Stupurienė, *gabriele.stupuriene@mii.vu.lt*
Vilnius University Institute of Data Science and Digital Technologies, Lithuania

Abstract
Constructionism as a learning paradigm is based on a design, actions and constructing things or solutions by using means via collaboration and the construction of knowledge (Papert & Harel, 1991). The main idea is that pupils can learn by performing activities by making things, or even deeper, by speaking about what they are performing (doing).

Our approach focus on informatics concept-based tasks (problems) as scaffolding activities to introduce informatics concepts to primary schools. The tasks are short and can help pupils to construct their knowledge and mental structures of informatics as a science discipline. In our long experience (more than ten years) hundreds of short tasks on informatics concepts were created.

Resent years many countries, and Lithuania among them, have been developing informatics (computer science, computing), or digital technologies, curriculum for primary school level. Usually the curriculum is based on integration with other subjects, nevertheless the principles and concepts of informatics are needed to be introduced. The key components and competence areas of informatics for primary school education needs to be identified and described.



Lithuania identifies the key components of informatics curriculum in 6 competence areas

Main educational goal is to provide the different teaching possibilities and environments supporting various activities and to motivate pupils to discover new ideas. Constructionist teaching and learning is one of the important approaches and takes place through conceptually open learning activities in individual as well as group exploration by seeking common knowledge and understanding.

Keywords
constructionism, informatics education, informatics concepts, short task, concept-based solving, primary education.

Design Science Research for Computational Thinking in Constructionist Education: A Pragmatistic Perspective

Vladimiras Dolgopolovas, *vladimiras.dolgopolovas@mii.vu.lt*
Valentina Dagienė, *valentina.dagiene@mii.vu.lt*
Eglė Jasutė, *egle.jasute@mii.vu.lt*
Tatjana Jevsikova, *tatjana.jevsikova@mii.vu.lt*
Vilnius University, **Institute of Data Science and Digital Technologies, Lithuania**

Abstract
Constructionist educational environments have provided solutions for an effective learning process. What are the criteria of the effectiveness of using such environments? The question is even more difficult if we are seeking computer enhanced constructionist solutions. One more problem is how to train the computational thinking (CT) skills and especially to develop their assessment. If we consider the level of understanding of the relevant scientific knowledge to be transferred to the learner as the effectiveness criteria, like done in such traditionally positivistic fields like physics or mathematics, then how does it relate to the computational thinking skills and abilities? This is an even more difficult question to answer in the case of multi-paradigmatic fields like engineering or informatics. The nature of knowledge in these disciplines is non-positivistic in general, since such research areas usually include not only technical but also social domains and the research methods in these fields are mainly qualitative.

The article makes an attempt to start a discussion on the above presented problems. This research aims at a detailed analysis and study of the possibility to arrange a constructionist platform and the relevant software enhanced educational environment in an effective way as it could be seen from the point of view of acquisition of CT skills. The possible solution is based on Design Science Research (DSR) methodology. The DSR methodology is developed under the pragmatistic research paradigm and could provide a framework for unification of educational solutions, especially in such educational fields like informatics and Science, Technology, Engineering, and Mathematics (STEM).

Research questions as related to software enhanced constructionist educational platforms are formulated as follows: (RQ1) What are the relations between inquiry-based learning (IBL) and CT? How can IBL help to develop CT? (RQ2) What are the advantages and possible disadvantages of inductive-deductive reasoning in connection with IBL? What is the role of abductive reasoning as related to IBL and CT? (RQ3) How to effectively (as related to CT) incorporate DSR to IBL? The article focuses on cognitive and epistemological aspects of the presented approach and analyzes the possible connection of DSR to the inquiry-based educational process. Another important aspect is the possible approach to IBL in the paradigmatically non-positivistic areas of scientific research. Pragmatistic aspects of DSR, based on the implementation of abductive and circumscriptive cognitive features, allow implementation of an inquiry-based educational process that focuses on CT in various inter-disciplinary areas like, for example, informatics.

Keywords
Computational thinking, constructionist education using technology, design science research, inquiry based education, pragmatism.

Computational Thinking in Teacher Professional Development Programs

Xiaoxue Du, *xd2164@tc.columbia.edu*
Columbia University, New York, NY, USA

Kay Chioma Igwe, *kci2104@columbia.edu*
Columbia University, New York, NY, USA

Abstract
The integration of computational thinking, CT, through technology, into curriculum strongly depend on teacher openness to compliance, of which is strongly correlated to awareness and technological skill level. Here, technological skill refers to ability to manipulate hardware, such as using problem solving to build a simple car robot, and/or software creation and implementation. In this pilot study, we create a project-based technology based workshop for teachers that introduce principles of computational thinking and analyze their abilities and thoughts about adding these learning tools into their curriculum. Six pre and in-service teachers in New York City were recruited to participate in this forty five minute workshop at separate times. The results suggest that technologically driven project-based learning design principles have the potential to increase participants' confidence and as a consequence motivate them to consider applying computational thinking related pedagogical practices in their classroom especially in subjects relating to math and science. Although CT pedagogical practice is rather technical and new to the majority of the participants, technology driven workshops may be necessary to increase teacher participation in CT classroom CT practices. This paper outlines the process by which computational thinking, through technology and problem solving, can be embedded in a professional training program for teachers. The results suggest that teachers need to have an understanding of technology and ways to implement computational thinking in order for it to be introduced in curriculum.

Keywords
computational thinking; project-based learning; teacher professional development; pedagogical practice; curriculum; workshop; program

Hardware and software environment used in this pilot study

What is Constructionism? Views from a Thai Perspective

Deborah A. Fields, *deborah.fields@usu.edu*
Utah State University, USA

Paulo Blikstein, *paulob@stanford.edu*
Stanford University, USA

Abstract
When creating and defining constructionism alongside with his colleagues, Seymour Papert refused to contradict his own principles by offering a set-in-stone definition of constructionism. Instead, he advocated for people to create their own definitions through the telling of deep narratives around learning. Intentionally or not he opened up a wide and complex set of possibilities for implementing and thinking about constructionism. But what happens when people outside of Papert's close circle of students and colleagues try to implement constructionism? What happens when people have indeed the license to appropriate it in very personal and culturally-aware ways?

This paper investigates one such case: the interpretation and implementation of constructionist principles in Thailand. In 1996 a group of MIT graduates from Thailand started a foundation to support what they hoped would be a transformative initiative to reshape education in their country. They brought Seymour Papert and later many other researchers from MIT, including his then student David Cavallo, to visit Thailand and begin a project to introduce constructionism to the country, starting first with teachers then reaching out through non-formal education to people in rural communities. After a few years, the collaboration with MIT phased out and the movement was left largely to its own devices, with relatively little contact with U.S. institutions or researchers. The foundation, led by a few visionary figures and several educators and leaders, with some support from the ministry of education and various corporations, continued to figure out on their own how to implement constructionism in Thailand. Intriguingly the Thai leadership and people who joined the constructionist community applied the philosophy not just in K-12 education but also in industry (from chemical companies to banks), remote farming villages, technical colleges, and non-formal education. Interviewing key founders of the movement as well as teachers, village leaders, business people, and others vested for many years in the constructionism movement in Thailand, we ask what constructionism means in Thailand within the diverse community in which it has been applied.

Keywords
constructionism, developing countries, educational reform, Thai perspective

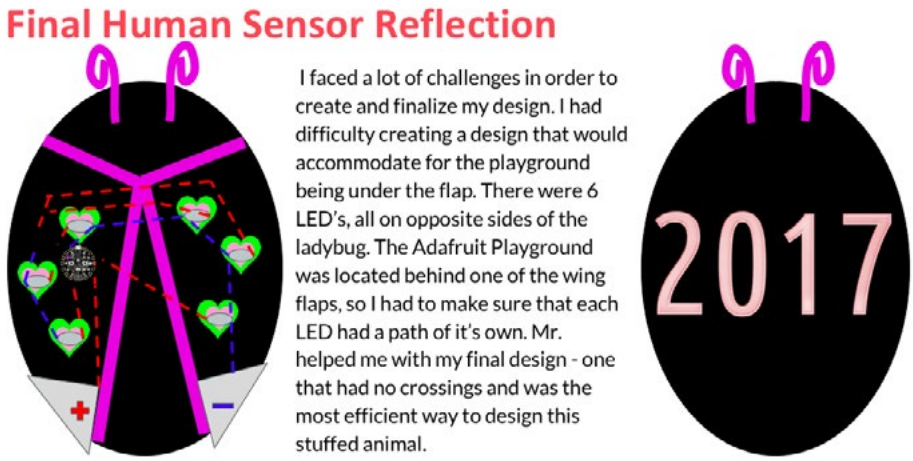
Personal Learning Journeys: Reflective Portfolios as "Objects-to-Learn-With" in an E-textiles High School Class

Deborah A. Fields, *deborah.fields@usu.edu*
Utah State University, Logan, UT, USA

Mia S. Shaw, *mshaw12@gse.upenn.edu*
University of Pennsylvania, Philadelphia, PA, USA

Yasmin B. Kafai, *kafai@upenn.edu*
University of Pennsylvania, Philadelphia, PA, USA

Abstract
Much attention in constructionism has focused on the design of learning tools and support for students building artifacts. Far less attention has been placed on reflection and reflective artifacts that let students consider their own learning. In this paper we share an analysis of portfolios in which high school students reflected on the design of their electronic textile projects during an eight-week curricular unit in an *Exploring Computer Science* class, an introductory computing course for high school (secondary) students in the United States. We examine portfolios as sites of student self-authorship: places where students showed agency in positioning themselves in relation to how they made their e-textile projects and to computer science more generally. In the discussion we consider the implications of reflective portfolios as "objects-to-learn-with" for educational implementation and constructionist pedagogy.



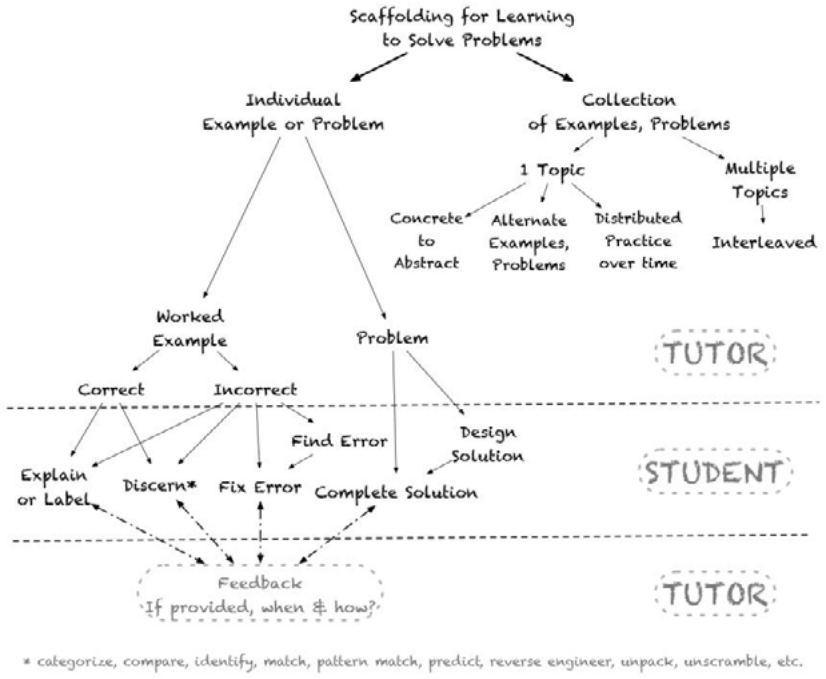
Ashley's portfolio page on challenges in the human sensor project.

Keywords
Electronic textiles, computer science education, identity, portfolios, objects-to-think-with

Constructionism and De-Constructionism as Complementary Pedagogies

Jean M. Griffin, *jean.griffin@temple.edu*
Temple University, USA

Abstract
Constructionism is advantageous for learners of all ages but underutilized in "serious" formal education settings. It can be challenging for teachers to allocate enough time for students to construct their own designs if the students are expected to master numerous concepts and skills. Also, some teachers lack experience with evaluating creative constructions. Students often want to abandon their designs when they encounter obstacles or bugs. Constructionism might be more widely adopted if teachers knew how to pair it with a pedagogy that ensures coverage of key concepts, provides effective practice with core skills, and helps students gain confidence with troubleshooting. This paper presents such a pedagogy, de-constructionism, a learning-by-taking-apart approach grounded in reverse engineering and practice theory. An experiment is described where students learning to program solve practice problems designed with a de-constructionist approach. Students' attitudes about the problems are reported. Suggestions for balancing constructionist activities with de-constructionist ones are discussed.



Model for De-Construction

Keywords
constructionism; De-Constructionism, code comprehension, bugs

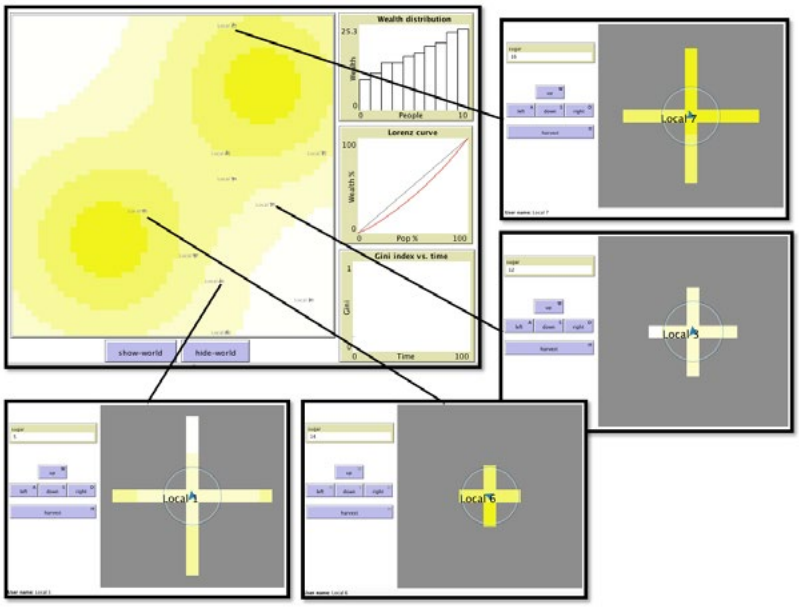
Mind the Gap: Teaching High School Students about Wealth Inequality through Agent-based Participatory Simulations

Yu Guo, *yuguo2012@u.northwestern.edu*
Northwestern University, USA

Uri Wilensky, *uri@northwestern.edu*
Northwestern University, USA

Abstract
This research paper presents a design-based research study centring on a constructionist curricular unit, called Mind the Gap (MTG), which was designed to help high school students learn about a complex and controversial social issue in the United States—wealth inequality. The four-day-long unit was implemented in eight economics classes with a total of more than 200 students across two high schools with vastly different demographics. In both schools, students' engagement with the unit was high. Preliminary data analysis has shown that students 1) made connections between their simulation experience and the real world to reason about wealth inequality, and 2) showed attitudinal changes favouring more equality after the unit.

MTG revolves around a series of three participatory simulations, which are microworlds that allow students to project themselves into through their own avatars like in a multiplayer online game and interact with the virtual environments in order to "figure out" the rules embedded in these simulations. This work contributes to both the literature of designing constructionist learning environment and people's perception and understanding of wealth inequality. In democratic countries, people's understanding of inequality is the key to achieve more equal societies.



Mind the Gap ABPS. Upper left: teacher's view; the other four: students' view

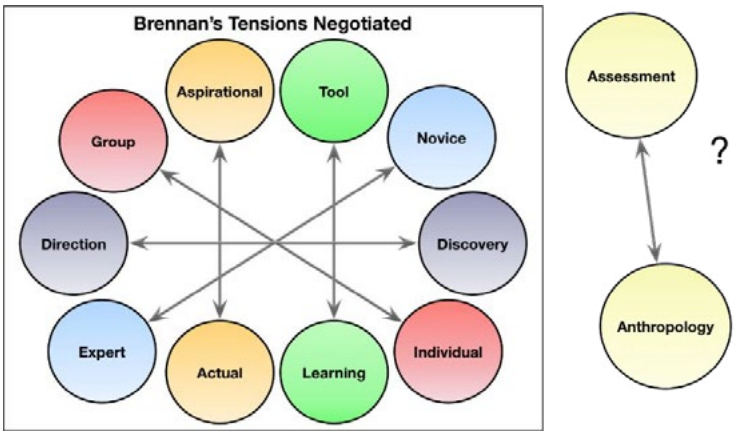
Keywords
Agent-based participatory simulation; wealth inequality; economics curriculum; complex social phenomena; artificial society

To Assess or Not to Assess: Tensions Negotiated in Six Years of Teaching Teachers about Computational Thinking

Daniel Hickmott, *Daniel.Hickmott@uon.edu*
School of Education, The University of Newcastle, Australia

Elena Prieto-Rodriguez, *Elena.Prieto@newcastle.edu.au*
School of Education, The University of Newcastle, Australia

Abstract
Coding and computational thinking have recently become compulsory skills in many school curricula around the world. This presents a major challenge for teachers, as many of them would not have had formal education on the underlying concepts needed to apply these skills, or the pedagogies to teach them. A range of professional development opportunities are currently being offered to teachers to address this challenge. One notable example of a model of PD designed to include constructionist learning experiences is *ScratchEd*, which was designed by Karen Brennan. Upon reflecting on her experiences designing and running *ScratchEd* for seven years, Brennan identified five tensions that professional development providers may need to negotiate, and proposed that these tensions could be used for scrutinising and critiquing professional development.



The five 'tensions negotiated' by Karen Brennan and our additional tension

In this paper, we analyse the process we have followed in the past six years to design, evaluate and improve our professional development through the lens of Brennan's 'tensions negotiated'. While we have experienced the same tensions, we contend that the extent to which we assess teachers' learning is a new tension that extends those identified by Brennan. There are strong reasons to assess teachers' knowledge, however, quantitative measures of learning could be at odds with Constructionism: as Papert argued in *Mindstorms*, constructionist educators should study their learning environments as anthropologists. Consequently, we have called this new tension the *tension between anthropology and assessment*.

Keywords
Teacher professional development; Constructionism; computational thinking; programming; pedagogical content knowledge.

Sharing is Caring in the Commons – Students' Conceptions about Sharing and Sustainability in Social-Ecological Systems

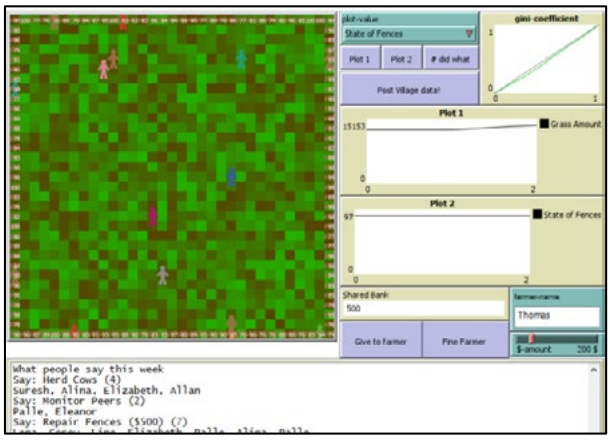
Arthur Hjorth, *arthur.hjorth@u.northwestern.edu*
Center for Connected Learning and Computer-Based Modeling, Northwestern University, USA

Corey Brady, *corey.brady@vanderbilt.edu*
Department of Teaching and Learning, Vanderbilt University, USA

Uri Wilensky, *uri@northwestern.edu*
Center for Connected Learning and Computer-Based Modeling, Northwestern University, USA

Abstract

In this paper, we present the design and analysis of a high school environmental science participatory simulation-based activity about sharing natural resources. In particular, we focus on this activity's utility in surfacing students' diverse ways of thinking about Social-Ecological Systems, offering entry points for Constructionist design. The activity was implemented as part of a three-week unit in an environmental science AP class. At the core of the activity was a participatory NetLogo simulation in which students played the role of dairy farmers. Through their interactions with the simulation and collective and individual decision-making, students struggled to reason and argue productively about the difficulties involved with sharing natural resource systems.



The virtual grazing ground and accompanying data interface gives students insights into how their individual and collective decisions affect the ecosystem.

We gave students two written assignments, in which they were asked questions about true, historical descriptions of communities sharing natural resource systems. Here we analyse the responses, identifying four distinct 'thinking patterns' across student responses. We discuss how these four patterns were productive in so far as they helped students to reason about the historical cases, but also how these patterns in their thinking restricted them from thinking productively about the full nature of the case studies. Finally, we discuss how future designs might address the less productive aspects of these patterns.

Keywords

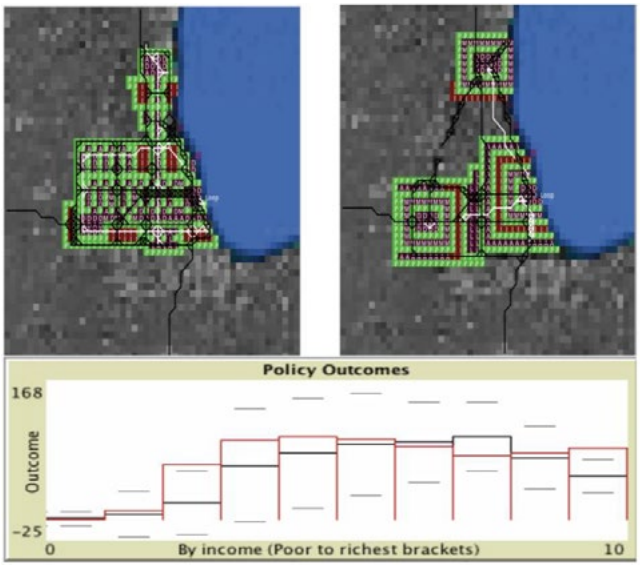
NetLogo, agent-based modelling, HubNet, participatory simulation, social studies, complex systems

Urban Planning-in-Pieces: A Computational Approach to Understanding Conceptual Change and Causal Reasoning about Urban Planning

Arthur Hjorth, *arthur.hjorth@u.northwestern.edu*
Center for Connected Learning and Computational Modeling, Northwestern University, USA

Uri Wilensky, *uri@northwestern.edu*
Center for Connected Learning and Computational Modeling, Northwestern University, USA

Abstract
In this paper we report on data collected in an Introduction to Social Policy class for Social Policy majors. We designed a Constructionist learning environment in which students collaboratively construct cities in order to optimize various policy outcomes. Our hope was that students would gain a better causal understanding of the effects of urban planning decisions. To investigate this, we analysed students' responses to pre- and post-questions, and looked for changes in their explanations of an urban planning-related question.



Sending and receiving common and private characters

More specifically, we use Association Rule Mining (ARM), a relatively simple Machine Learning approach designed to find co-occurrences in data sets. Drawing on the Knowledge Analysis (KA) framework, we use ARM to look at differences in how students assemble their explanations from smaller "explanation-pieces." We identify and discuss pre-post changes in two association rules, and argue that ARM is a particularly suitable approach to analysing changes in students' *causal* thinking in a KA-context.

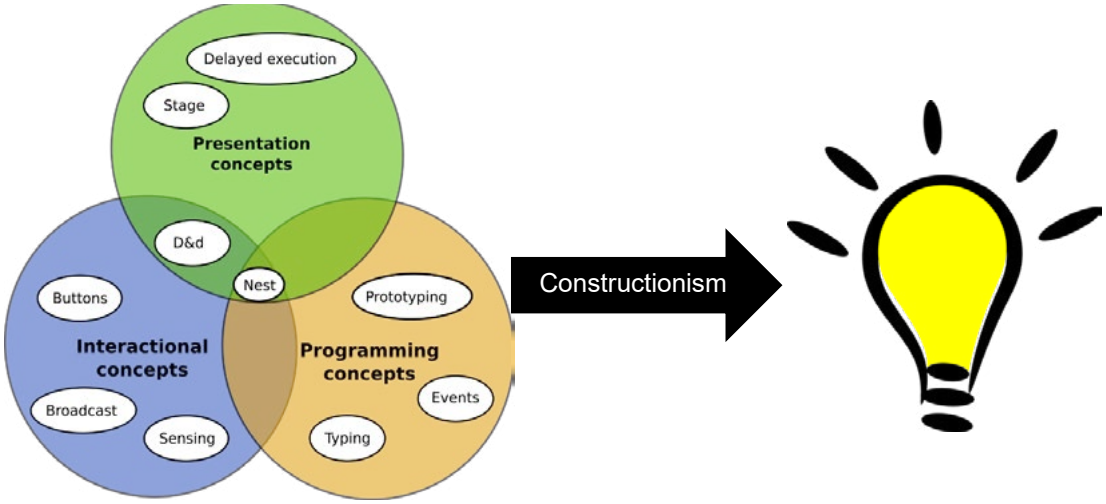
Keywords
knowledge analysis, netlogo, social studies, complex systems, causal reasoning, computational analysis, computational methods

How High is the Ceiling? Applying Core Concepts of Block-based Languages to Extend Programming Environments

Sven Jatzlau, *sven.jatzlau@fau.de*
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Ralf Romeike, *ralf.romeike@fau.de*
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Abstract
Since the emergence of block-based visual programming languages, they have been developed and improved to become increasingly accessible, intuitive, and easy to use. Over the course of this evolution, both uncommon and entirely new language concepts have been introduced, such as the cloning of objects, or the nesting of sprites. This paper provides a selection of core concepts and describes a categorization model. It is proposed that the concepts found in block-based languages are the reason they lend themselves to constructionist learning approaches. To illustrate this point, the fundamental computer science concept of image nesting will be selected, its background and origin explained, and its incarnation in current block-based languages outlined. Following this, the constructionist task of modifying a programming environment will be implemented. Using the aforementioned concept proves that even high-level solutions according to the "high ceiling"-principle can be implemented using the basic concepts of block-based languages.



Core concepts in block-based languages enable constructionist solutions to users' problems.

Keywords
visual programming languages; Snap; Scratch; GP; block-based languages; language concepts; nesting; composite objects

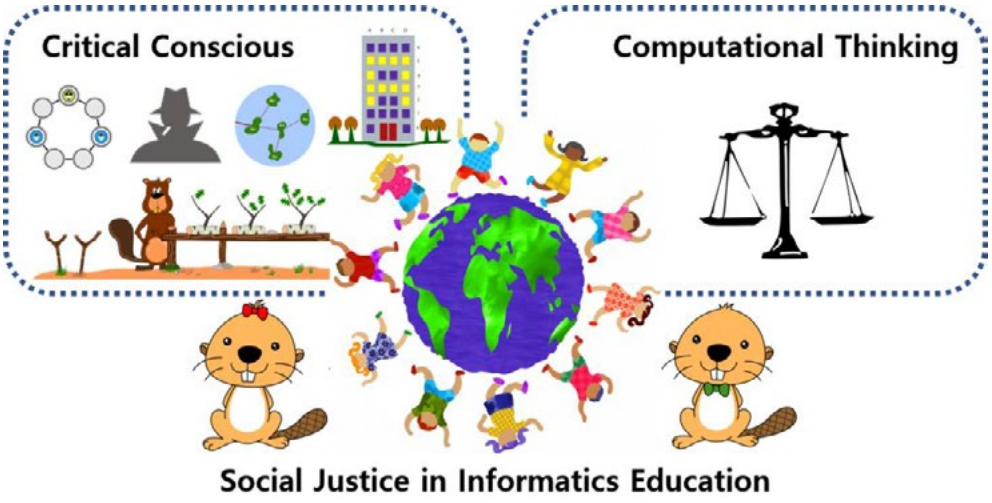
The Direction and Possibility for Social Justice in Informatics Education based on Bebras Challenge in Republic of Korea

Ungyeol Jung, *purnagi@gmail.com*
Korea National University of Education, Republic of Korea

Young-jun Lee, *yjlee@knue.ac.kr*
Korea National University of Education, Republic of Korea

Abstract
In addition to high interest in computational thinking ability, the 2015 revised national curriculum of Republic of Korea has set as an emphasis on revising skills for the 4th Industrial Revolution era through the enhancement of the ability, and has made informatics mandatory for the middle school. As a result, regional, gender, teacher level, and students' prior knowledge levels are pointed out as important environmental variables affecting informatics education and raise of students' computational thinking of computing, and as a threat to the social justice of informatics education.

In this study, based on the tasks and the results of the Bebras Challenge, the applicability of Bebras challenge for social justice in informatics education was analyzed from two perspectives (internal and external) and suggested implications. As a result of the research, the tasks of the Bebras challenge reflected educational elements for social justice. Also, the results of the Bebras contest showed that informatics education based on Bebras challenge could contribute to the social justice.



The directions and possibility for social justice in informatics education based on Bebras challenge

The results of this study will provide the basis for the direction and possibility of Bebras challenge for social justice in informatics education.

Keywords
Informatics Education, Software Education, Social Justice, Bebras Challenge

Interconnection between Computational Thinking and Digital Competence

Anita Juškevičienė, *anita.juskeviciene@mii.vu.lt*
Vilnius University, Lithuania

Valentina Dagienė, *valentina.dagiene@mii.vu.lt*
Vilnius University, Lithuania

Abstract
The European Commission Science Hub has been promoting Computational Thinking (CT) term as an important 21st century skill or competence. However „despite the high interest in developing computational thinking among schoolchildren and the large public and private investment in CT initiatives, there are a number of issues and challenges for the integration of CT in the school curricula[2]. " From the other side, the Digital Competence (DC) Framework 2.0 (DigCom) is promoted in the same European Commission Science Hub portal[3]. It shows that both topics have may things in common. Thus there is the need of research on CT relationship with digital competence.

The goal of this paper is to analyse and discuss the relationship between DC and CT, and help educators as well as educational policy makers to make informed decisions about how CT and DC can be included in their local institutions. We begin by defining DC and CT and then discuss the current state of both phenomena in education in multiple countries in Europe. By analysing official documents, we try to find the underlying commonness in both DC and CT, and discover all possible connections between them. Possible interconnections between both approaches components groups is presented in Fig. 1.

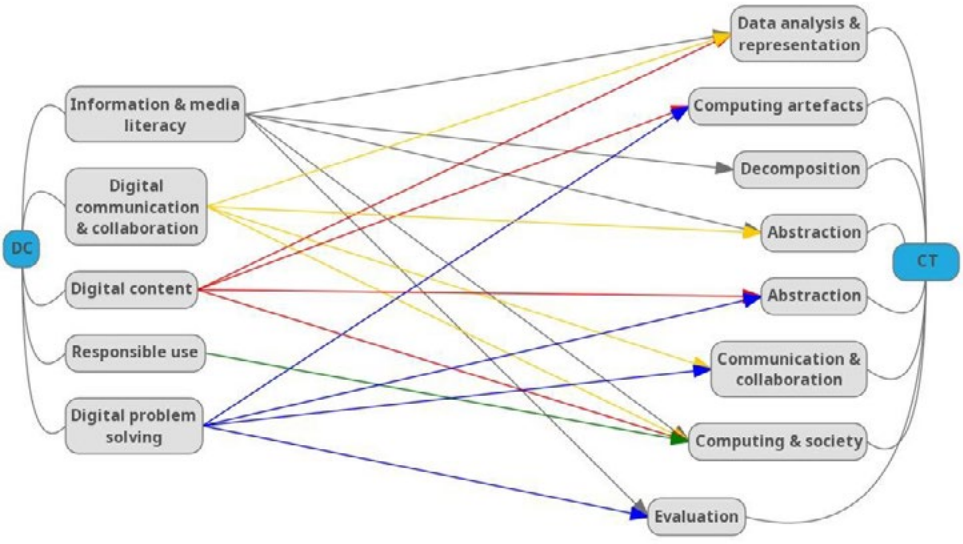


Fig. 1. The interconnections between DC and CT

Keywords
Computational thinking, digital competence, education

[2] https://ec.europa.eu/jrc/en/computational-thinking
[3] https://ec.europa.eu/jrc/en/digcomp/digital-competence-framework

AI Programming by Children

Ken Kahn, *toontalk@gmail.com*
Department of Education, University of Oxford, UK

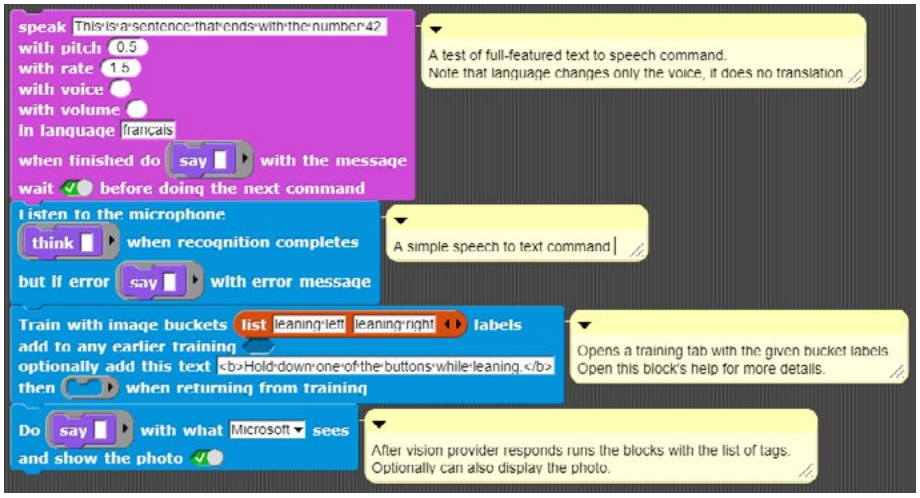
Niall Winters, *niall.winters@education.ox.ac.uk*
Department of Education, University of Oxford, UK

Abstract

The idea of children constructing artificial intelligence programs goes back to the early days of Logo (Papert & Solomon 1971; Kahn 1975; Kahn 1977). After decades of little activity recent efforts to support students in making AI programs has come from Stephen Wolfram (Wolfram 2017b), Google (Google 2018a, 2018b), the Machine Learning for Kids website (Lane 2018), and the eCraft2Learn project (eCraft2Learn 2018b). Two technological developments underlie the feasibility of these efforts: (1) AI cloud services and (2) mainstream laptops and desktop computers that now can run sophisticated machine learning algorithms. And all of these developments can be made accessible in a web browser, thereby running on many platforms without the need to install software.

Given appropriate programming tools children can make apps and intelligent robots that rely upon speech and image recognition services. They can add custom capabilities to their programs by using machine learning training and prediction. In doing so they may learn about perception, language, psychology, and the latest empowering technologies.

We describe the addition of new programming blocks to the Snap! visual programming language (Harvey & Mönig 2010) that provide easy-to-use interfaces to both AI cloud services and deep learning functionality. Interactive learning materials have been developed and preliminarily trialled with students. We anticipate in future trials to observe children creatively using these new blocks to build very impressive programs. Children are likely to be even more motivated to program when the results are such capable programs.



A small sample of new Snap! blocks for AI

Keywords

visual programming; machine learning; block languages; Snap!, AI services; Cloud services; speech synthesis; speech recognition; image recognition

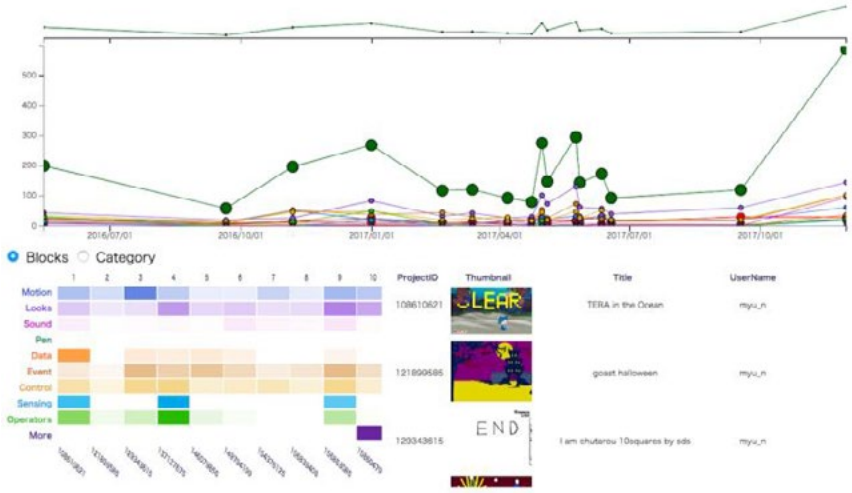
VISURATCH: Visualization Tool for Finding Characteristics of Teaching and Learning Process of Scratch Programmers

Nobuko Kishi, *kishi@tsuda.ac.jp*,
Mari Yoshida,
Minori Yoshizawa,
Tsuda University, Japan

Aoi Yoshida, *aoi@si.aoyama.ac.jp*
Aoyama Gakuin University, Japan

Abstract
We are developing an interactive visualization tool, VISURATCH, for Scratch projects using Python and D3.js. Our tool collects a set of programs, or projects, from the Scratch site (http://scratch.mit.edu) and visualizes each program's statistics, such as the number of blocks, in the chronological order of program creation or modification. With this tool, we aim to enhance an instructor or learner's ability to quickly assess the characteristics or the status of their Scratch programming knowledge.

VISURATCH's data visualization model is based on the Visual Information-Seeking Mantra: Overview first, zoom and filter, then details-on-demand. The figure below shows the main screen layout. The line chart in the upper part shows the number of blocks in each Scratch programming block category in chronological order. Scratch programs are built with a variety of block types, which are grouped into ten categories: *Motion, Looks, Sound, Pen, Data, Events, Control, Sensing, Operators,* and *More*. The density map or heat map on the bottom part shows the block usage of a subset of the programs selected in the line chart. A column corresponds to one Scratch program, and each square in the column shows the number of blocks in each block category with the darkness of its color.



Main screen layout of VISURATCH.

We find the tool shows some characteristics of a teaching plan when it is applied to a set of sample programs shared in a Scratch studio. We also find the tool shows some characteristics of a learner, such as the pattern of programming blocks frequently used, when applied to a set of programs specified by the learner's id.

Keywords
Information Visualization, Scratch Programming, Computer Science Education

Curriculum Intervention for Learning Programming in Python with Turtle Geometry

Eva Klimeková, *klimekova@fmhp.uniba.sk*
Department of Informatics Education, Comenius University, Bratislava, Slovakia

Abstract
Programming as part of computing education at upper secondary schools in Slovakia has a long tradition. When a new trend of teaching programming in Python emerged, many schools adopted it. In this paper we focus on deeper understanding of teaching programming basics in the Python language. Our aim is to find out how we can help to teacher for easier transformation to new language. We designed a curriculum intervention for learning programming in Python in which we decided to exploit the motivation of turtle geometry. This supports learning by exploring, creating and discovering. By means of it pupils learn the basic concepts and constructs as variables, loops, functions and conditions. We developed, implemented and verified our materials iteratively, using design research. In this paper we also present the process of development of our curriculum intervention from the first steps of collecting initial insight to implementing the materials into practice at upper secondary schools in Slovakia.

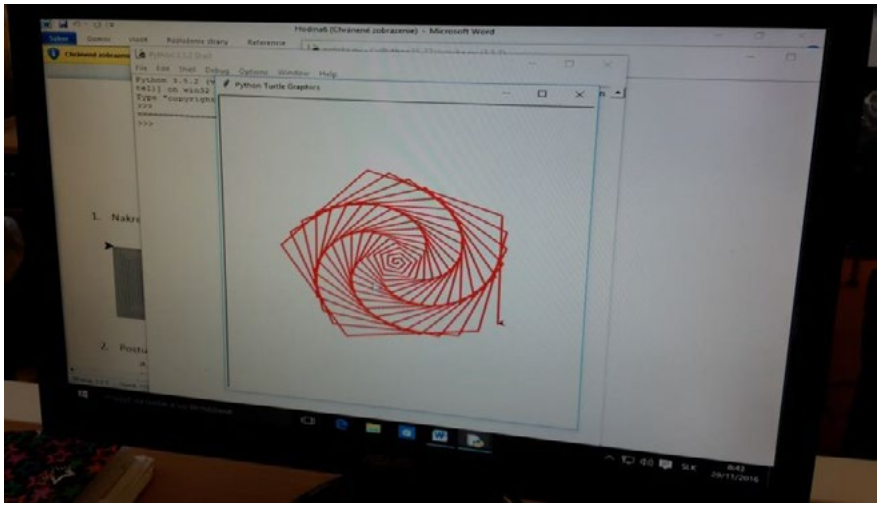


Photo of a pupil's "rose" created by using turtle geometry in Python

Keywords
Python, turtle geometry, curriculum intervention

The Impact and Effectiveness of Technology Enhanced Mathematics Learning

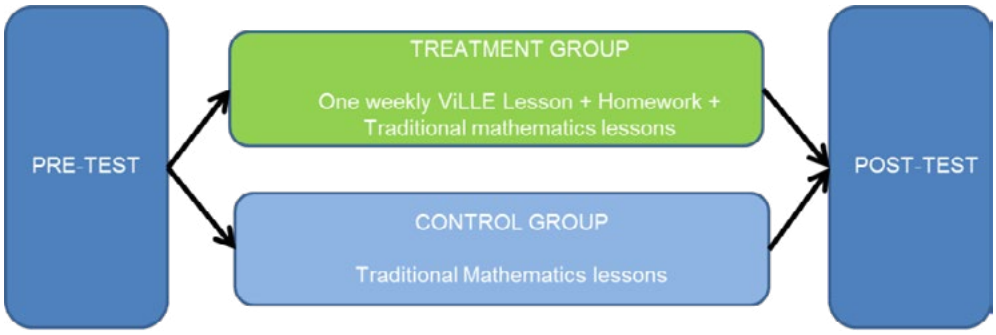
Einari Kurvinen, *emakur@utu.fi*
University of Turku, Finland

Valentina Dagienė, *valentina.dagiene@mii.vu.lt*
Vilnius University, Lithuania

Mikko-Jussi Laakso, *milaak@utu.fi*
University of Turku, Finland

Abstract
What happens inside the classroom is crucial for pupils' learning. Mathematics is mostly hard work and only small portion is accounted by talent. However emotions and enjoyment can facilitate the hard work and make it easily done. We can use technology to help pupils to complete more exercise, make practicing more interesting and motivate pupils, which in turn will improve learning performance and learning results.

A 15-week research study was conducted by research team of University of Turku, Finland and Vilnius University, Lithuania. Three schools were selected for this study. From each school we had two 3rd grade classes, one with control group and one treatment group. There was altogether 140 pupils (N=140) of which 69 (N=69) formed the treatment group and 71 (N=71) formed the control group. The figure below describes the research setup. The test started with a pre-test and was followed by 15 week research phase. The research was concluded with a post-test. During the research period, pupils from treatment group had one mathematics lesson per week transformed into technology enhanced mathematics lesson using a virtual learning platform called ViLLE. The control group followed typical teaching without any changes.



Research setup

Identical tests for pre-test and post-test were used in the study. The test consisted of two parts. The first part measured mathematics performance in topics taught in 3rd grade. The second part was an arithmetic fluency test that measures how fast pupils can solve basic arithmetic facts. The study clearly showed that the treatment group outperformed the control group pupils with statistical significant difference both in mathematics performance and in arithmetic fluency. When we combine these results with our previous findings there is solid evidence that the digital learning path in ViLLE makes a radical improvement to pupils' math learning. What makes it more amazing that we only replaced one lesson in a week.

Keywords
Gamification; Mathematics; Primary education; Technology enhanced learning; TEL

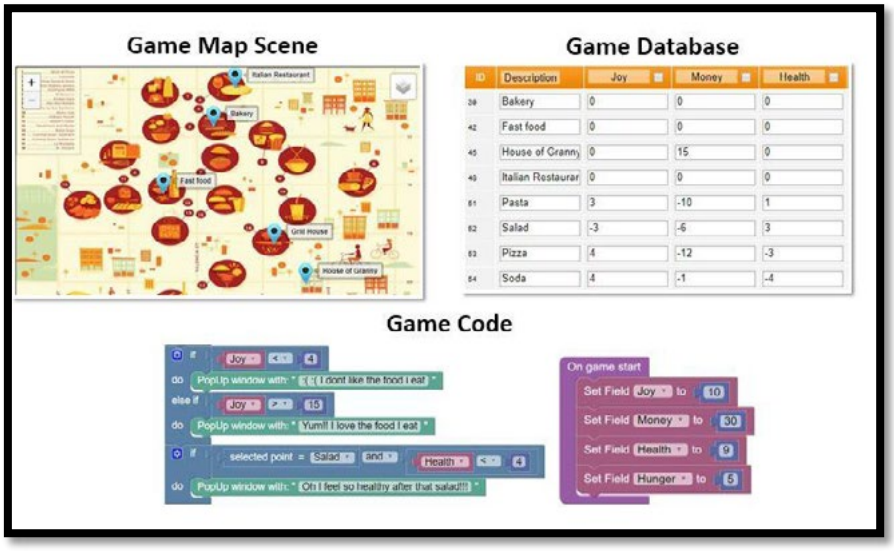
Constructionist Approaches to Computational Thinking: A Case of Game Modding with ChoiCo

Marianthi Grizioti, *mgriziot@ppp.uoa.gr*
National and Kapodistrian University of Athens, Educational Technology Lab, School of Philosophy, Faculty of Pedagogy, Greece

Chronis Kynigos, *kynigos@ppp.uoa.gr*
National and Kapodistrian University of Athens, Educational Technology Lab, School of Philosophy, Faculty of Pedagogy, Greece

Abstract
Computational Thinking (or CT) involves a wide range of mental processes like problem solving, recursive thinking, abstract thinking etc, which are considered necessary supplies for the 21st century children. However, despite the wide attention that CT has received, there is still limited research on pedagogical designs and strategies that can promote the acquisition and development of such skills. In this paper, we discuss how constructionist approaches can inform and benefit the cultivation of computational thinking by exploring the case of modifying and sharing digital games. In this context game modding is implemented as a constructionist activity that enables students' progressive engagement in computational thinking through their interaction with various affordances such as coding, data processors, graphics editors, etc. To further investigate this approach, we present the results of a design-based research in which junior-high school students modified games with the digital tool ChoiCo. ChoiCo (Choices with Consequences) in an online platform that integrates three different affordances for designing and modding digital games: a map-based (GIS) game scene, a simplified database and block-based programming editors. The research focused on a) the computational thinking skills that emerge and the meanings that are generated through students' engagement with the three affordances and b) how the progressive engagement in game modding can support the development of such skills.

Keywords
Computational thinking, game modding, block-based programming, progressive engagement



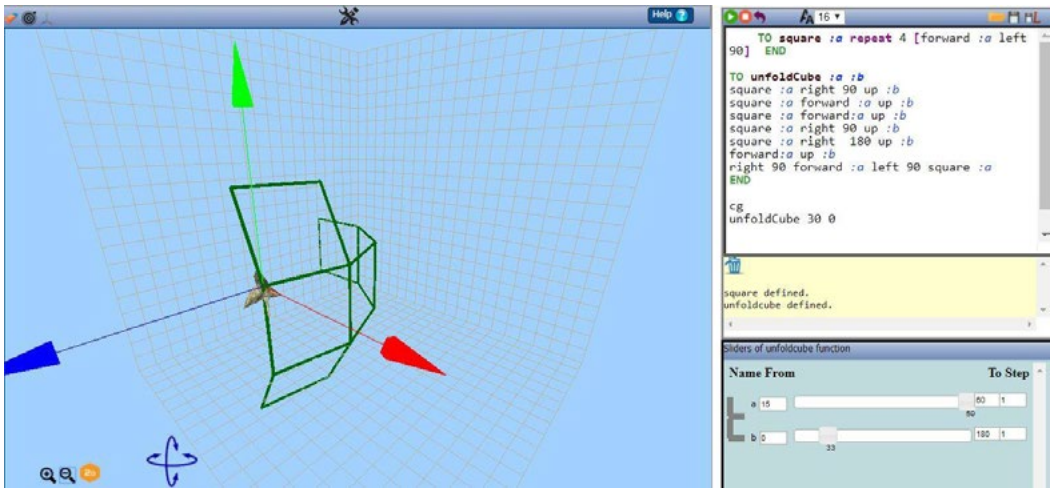
The three game design affordances of ChoiCo tool

Programming Approaches to Computational Thinking: Integrating Turtle Geometry, Dynamic Manipulation and 3D Space

Marianthi Grizioti, *mgriziot@ppp.uoa.gr*
National and Kapodistrian University of Athens Educational Technology Lab, School of Philosophy, Faculty of Pedagogy, Greece

Chronis Kynigos, *kynigos@ppp.uoa.gr*
National and Kapodistrian University of Athens Educational Technology Lab, School of Philosophy, Faculty of Pedagogy, Greece

Abstract
During the last decade, coding has come to the foreground of educational trends as a strong mean for developing students' Computational Thinking (or CT). However, there is still limited research that looks at coding and Computational Thinking activities through the lens of constructionism. In this paper, we discuss how the knowledge we already have from other thinking paradigms and pedagogical theories, such as constructionism and mathematical thinking, can inform new integrated designs for the cultivation of Computational Thinking. In this context, we explore students' engagement with MaLT (Machine Lab Turtle-sphere)[4], an online environment of our design that integrates Logo textual programming with the affordances of dynamic manipulation, 3D graphics and camera navigation. We also present a study on how the integration of the above affordances can promote constructionist learning and lead to the development of CT skills along with the generation of meanings about programming concepts.



The MaLT environment that integrates Logo programming with 3D graphics, dynamic manipulation and camera navigation

Keywords
Logo Geometry; Computational Thinking; Programming; Dynamic Manipulation; 3D graphics

[4] http://etl.ppp.uoa.gr/malt2/

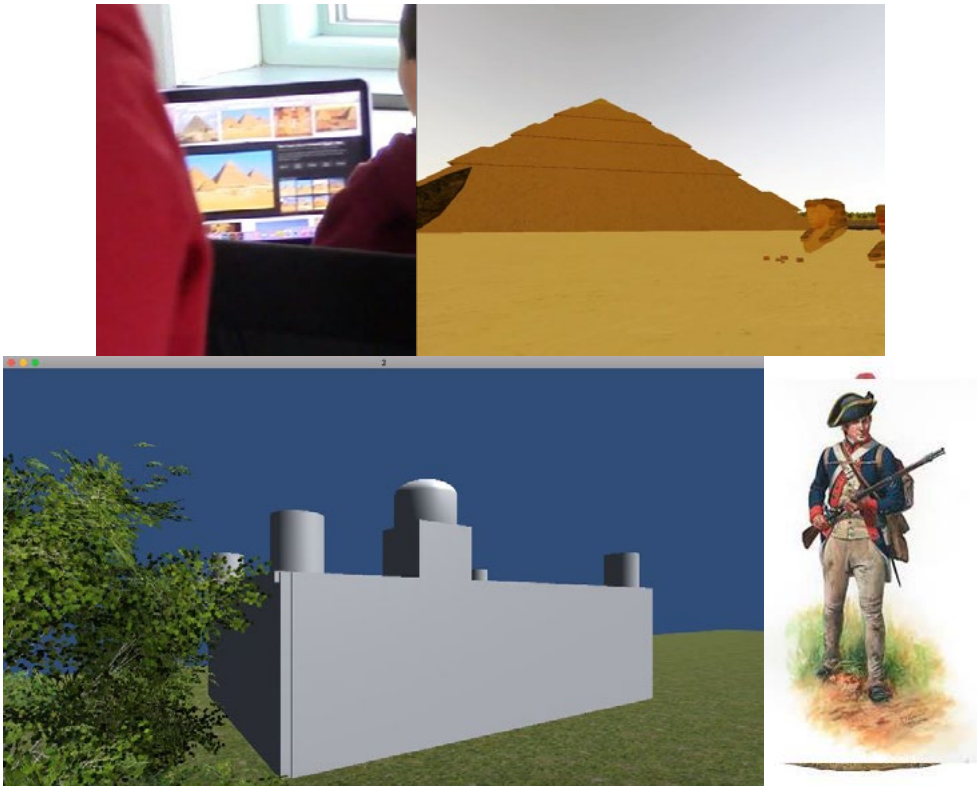
Modeling Time

Kit Martin, *kitmartin@u.northwestern.edu*
Northwestern University, USA

Gabriella Anton, *gabriellaanton3.2020@u.northwestern.edu*
Northwestern University, USA

Abstract

We held a two-day workshop where eight 5th graders constructed historical simulations in Unity. We situate the work in the theoretical positions of constructionist history, figured worlds, and narratives. We find that students could use Unity to make figured worlds, discussing their construction and the origins of the sources they used to make the environments. This constructionist learning environment highlighted aesthetics and collaboration. When children played the game of "what if I was traveling to the past," they constructed representations of the past. They assigned new meanings to the digital authoring space available in Unity to turn it into ancient Egypt, the Stone Age, ancient Mali, the future, and Revolutionary War America. They chose to go there in their play. This is very much like how Vygotsky saw students turn the area behind the couch into the robbers' den. They used collectively developed signs to detach themselves and enter an imagined world of play.



5th graders built historical sites including Egypt, the ice age, the Revolutionary War, the Jurassic and future. Some students referred to image sources.

Keywords

History, Constructionism, Unity, Learning Environment

Ant Adaptation: A Complex Interactive Multitouch Game about Ants Designed for Museums

Kit Martin, *kitmartin@u.northwestern.edu*
Northwestern University, USA

Michael Horn, *michael-horn@northwestern.edu*
Northwestern University, USA

Uri Wilensky, *uri@northwestern.edu*
Northwestern University, USA

Abstract
Leaf-cutter ants in South and Central America are the major herbivores in the environment. Their trails range for several hundred meters from their colony. Long lines of ants with shreds of leaves and flowers, sometimes up to four times the weight of their tiny bodies, march from the forest back to their subterranean nests. For a long time, myrmecologists could only guess what the leaf-cutters were doing with all that roughage. The scientists observed ants in the field and they found something astounding. Inside every leaf-cutter ant nest, myrmecologists discovered chambers of fungi. The fungi are tended by pygmy ants. These pygmies––like coal men working in the depths of a 19th century steamer––take the leaves foraged by the larger terrestrial worker ants and pile them at the base of each fungus. The fungi in turn extract nutrients from the decomposing leafy material. They then use the nutrients from the process to make spores. The ants then, after feeding the fungi, collect the fungi's spores, which they in turn eat. The ants are farmers.

Such a discovery could only be made by close observation and long-term desire to answer the question, Where does all the roughage go? Kit has been watching ants for a long time now, asking similar questions. When he was seven, living in Kenya, he stood next to a Safari ant trail for two hours, just watching them. Safari ants do not have nests. Instead, they march in long lines, foraging for food as they go. This paper describes the design of, and visitor interaction with, an interactive tabletop game that brings this experience into the hands of a seven-year-old in a large midwestern natural history museum. During this pilot, we observed visitors play the game and talked to them about the experience. To demonstrate our analysis and the preliminary lessons learned, we describe units of knowledge one 7-year-old developed through play.

Luck and skill weave the action of an emergent system of ants into the players' attempt to beat their opponents as they swipe and touch interacted with an ant microworld.

Keywords
keyword; ants; complex systems; informal learning; constructionist epistemology; methodologies; tools and technologies; innovative computing education

Enabling Collaboration and Tinkering: A Version Control System for Block-based Languages

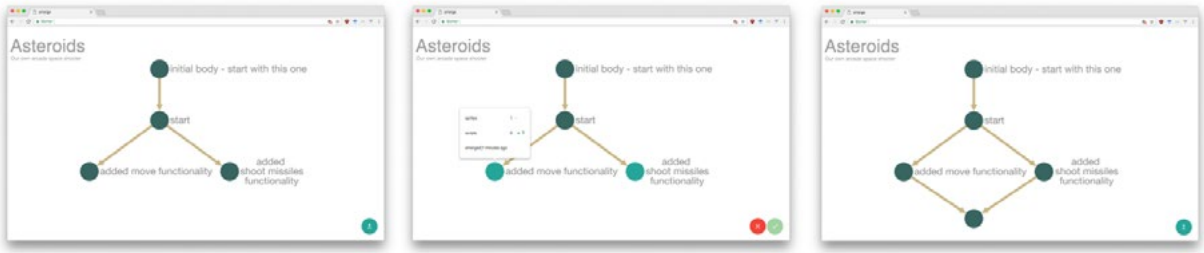
Tilman Michaeli*, *tilman.michaeli@fau.de*
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Stefan Seegerer*, *stefan.seegerer@fau.de*
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Ralf Romeike, *ralf.romeike@fau.de*
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Abstract

Version control systems are essential for coordinating teamwork when working in projects. They support computational thinking practices such as collaboration and tinkering. Yet, when using block-based languages, which are an excellent choice for novice programmers, there is no adequate solution that allows this form of collaboration. This paper presents a concept for a simple and easy-to-use web-based version control system as well as an exemplary implementation for the popular language *Snap!*. Based on an analysis of existing version control systems, their use in Computer Science Education in university and school contexts, and specifics of block-based-languages, key design principles for a version control system for block-based languages are outlined. Based on this, possible use cases for such a version control system in classroom environments will be discussed.



Easy-to-use version control system for collaboration and tinkering with block-based languages

Keywords

Version Control System; Block-based Languages; Snap!; Computational Thinking; Collaboration; Tinkering

* These authors contributed equally to this work.

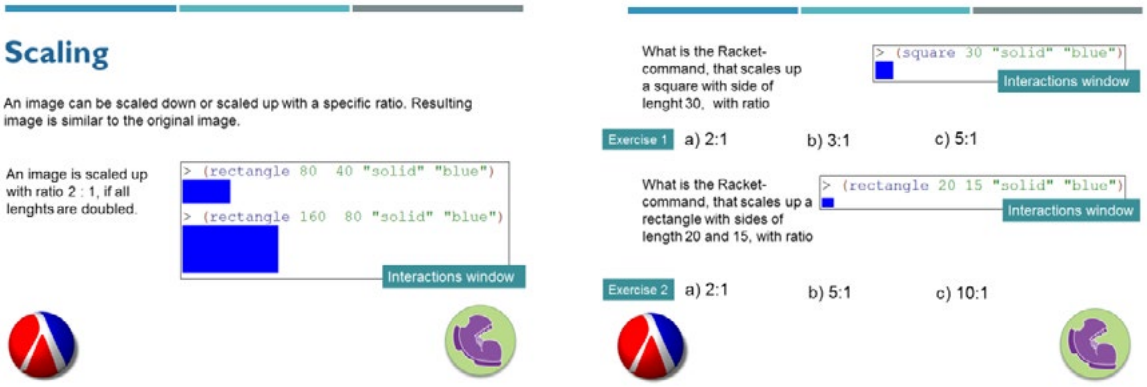
Racket Programming Material for Finnish Elementary Math Education

Tiina Partanen, *tiina.s.partanen@tampere.fi*
City of Tampere, Finland

Pia Niemelä, *pia.niemela@tut.fi*
Tampere University of Technology, Finland

Timo Poranen, *timo.t.poranen@uta.fi*
University of Tampere, Finland

Abstract
Programming, more widely perceived as computational thinking, is becoming a part of basic education in many countries. 2014 Finnish National Curriculum (FNC-2014) for K-9 education came into effect in 2016, and it embeds programming in the subjects of mathematics and crafts. The change in the curriculum challenges teacher training and causes a demand for appropriate teaching materials. This paper introduces a free Racket programming material for grades 5 to 9 developed by the authors to be fully integrable with mathematics lessons. Racket is a general purpose multi-paradigm programming language with features especially fit for computer science education.



Theory and exercise material on scaling rectangles in grades 5 and 6

The material consists of theory sections, corresponding exercises and answers. Covered math topics comprise natural and rational numbers, arithmetic operations, the order of operations, rounding, percentage calculations, scaling, polynomials, truth values, coordinate systems, geometrical objects, angles, etc. We evaluate the math suitability of material by comparing the introduced concepts with the ones in the FNC-2014 mathematics syllabus. In addition, the feedback received during teacher training is exploited in evaluation. This study follows the principles of educational design research, where the feedback is used for incremental improvements. The underlying research questions are: i) Which kind of programming theory, material and exercises are foreseen beneficial for learning mathematics? and ii) How do the introduced exercises comply with FNC-2014 and its underlying learning theory?

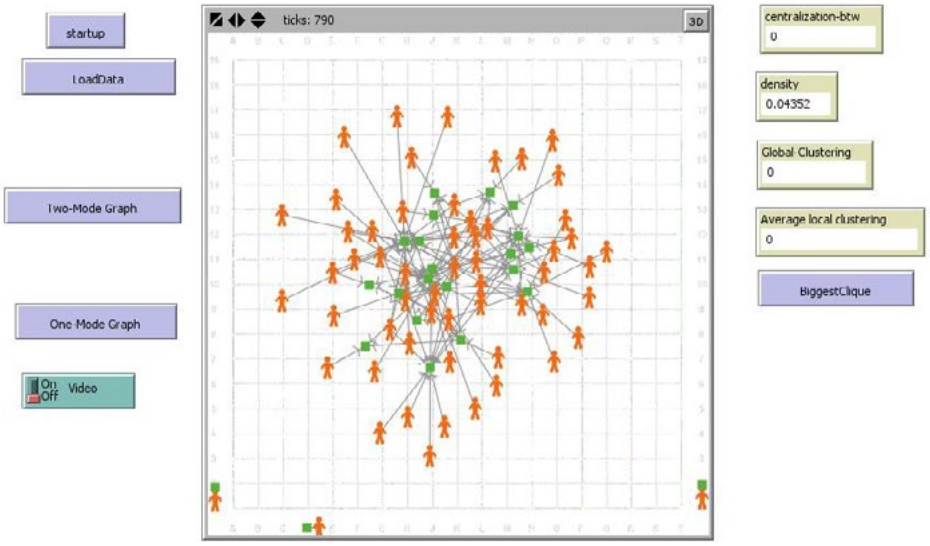
Keywords
Racket programming material, computational thinking, computer science education, spiral curriculum, grades 5-9, mathematics

Using Agent-based Modelling of Collaboration for Social Reflection

Evgeny Patarakin, *patarakined@mgpu.ru*
Moscow City University, Russia

Abstract

Introducing students and teachers to the network science may begin with mapping their own activities in the networked communities. The advantage of this approach is a network lens is used to understand situations in which students and teachers are drawn into. In different learning communities there is a definite "virtual chips" over which agents perform their actions. This simplest model for sociograms creation based on the actions with social objects can be applied to various collaboration situations. In all cases when actors perform actions over shareable objects, the log for these actions can help build sociograms as well as conduct social research.



Interface of Dynamic Wikigram Model

We start from the hypothesis that social reflection can be triggered by sociograms and we argue that learning as `building knowledge structures' happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, that can be discussed, evaluated and reused by other participants to create new entities, and data on the interactions of agents of learning can be presented as a map.

Keywords

Collaboration; ABM; Netlogo; wiki; social reflection

Reconstructing Constructionism by Construal

Nicolas Pope, *nwpope@gmail.com*
University of Turku, Finland

Jonathan Foss, *jonathan.foss@warwick.ac.uk*
University of Warwick, Coventry, CV4 7AL, UK

Meurig Beynon, *wmb@dcs.warwick.ac.uk*
University of Warwick, Coventry, CV4 7AL, UK

Abstract
Noss and Clayson (2015) sets out an agenda for 'reconstructing constructionism', identifying as a major challenge for research: "... to transform constructionism from a framework for action into a set of ways of conceptualising what people do in constructionist environments that can simultaneously assist in designing those environments." This paper relates this challenge to an approach to constructionism based on 'making construals' within the alternative conceptual framework for computing of Empirical Modelling (EM). Making construals is a digital skill based on three fundamental concepts: observables, dependency and agency ("ODA"). Its application will be illustrated using the online Construit environment. Making construals promotes an empirical and experiential perspective on computing that is complementary to 'computational thinking'. Its epistemological roots reflect the deep appreciation of the role of personal experience that characterises the pragmatism of William James and John Dewey.

This paper reviews and illustrates an EM approach to constructionism with respect to six topics: provocative modelling, programming paradigms, objects-to-think-with, definitive programming, Scratch as a 'broader constructionist methodology', and emerging empirical perspectives on computing and learning. The progression of topics relates an EM approach to ideas, tools and practices to which current trends in computer support for constructionism are converging.

Jonathan Foss's construal of the Artiphon: learning music theory through construction

The concluding section considers a central problem, articulated by diSessa and Cobb (2004) and cited by Noss and Clayson, concerning constructionism as a framework for action where theories "are relatively inexplicit, complex, and often involve multiple very diverse elements that cannot plausibly be brought under a single umbrella." It argues that a pragmatic approach to constructionism can dissolve this problem, bringing unity to the plurality of approaches to computing and learning it involves.

Keywords
constructionism; objects-to-think-with; making construals; Empirical Modelling; radical empiricism

Constructionist STEM Activities Using the Bridge21 Model

Brendan Tangney, *tangney@tcd.ie*
The Trinity Centre for Research in IT in Education, School of Computer Science & Statistics and School of Education, Trinity College Dublin, the University of Dublin, Dublin 2, Ireland

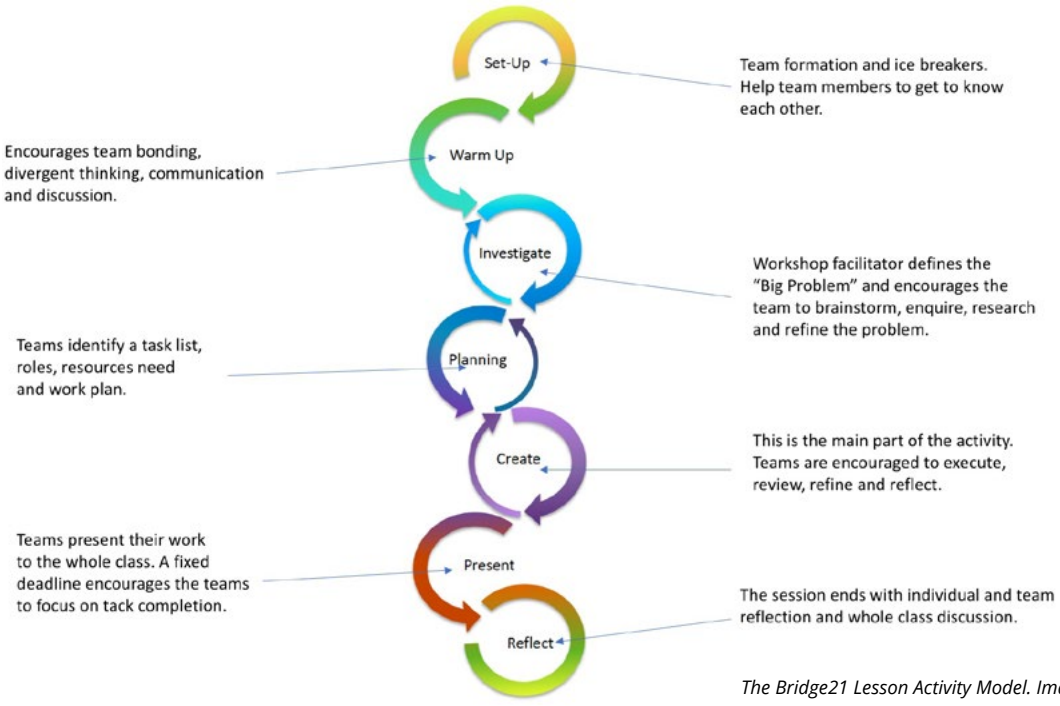
Ian Boran, *ian.boran@tallaghtcs.ie*
Tallaght Community School, Dublin 24, Ireland

Tony Knox, *tonyknox@kilkennycollege.ie*
Kilkenny College, Kilkenny, Ireland

Aibhín Bray, *brayai@tcd.ie*
The Trinity Centre for Research in IT in Education, School of Computer Science & Statistics and School of Education, Trinity College Dublin, the University of Dublin, Dublin 2, Ireland

Abstract

This paper explores how the use of a particular model of 21st Century teaching and learning (Bridge21), which aligns easily with a constructionist approach, can be integrated into an authentic classroom context, and the extent to which this leads to increases in students' metacognitive problem-solving skills as well as their engagement and confidence with STEM disciplines. The research is framed as an exploratory case study with two embedded units: the first focused on developing problem-solving skills in a group of 21 students (15-16 years old) in an after-school STEM club, using technologies that included a microworld for the construction of bridges; the second focused on the teaching of mathematical functions using a graphics calculator, with a view to improving student confidence and level of mathematical engagement, in a class of 24 students (15-17 year old) of mixed gender and weak to modest mathematical ability. Statistically significant improvements in students' metacognitive problem solving skills are reported from Intervention A, while Intervention B reports statistically significant improvements in students' attitude to mathematics and technology.



The Bridge21 Lesson Activity Model. Image adapted from (Byrne, O'Sullivan & Sullivan 2017).

Keywords

STEM; 21st Century Teaching and Learning; Bridge21; Constructionism

Exploring Girls' Values and Perspectives in Making for Others

Sawaros Thanapornsangsuth, *st2839@tc.columbia.edu*
Teachers College, Columbia University, USA

Nathan Holbert, *holbert@tc.columbia.edu*
Teachers College, Columbia University, USA

Abstract
This study explores a pedagogical framework on how "making for others" can influence and engage girls in maker activities. We are particularly interested in how the relationship between builders and their clients can influence the builders' making process and their motivation throughout that process. Feminist literature suggests that girls tend to locate themselves in relation to the world and describe themselves through actions that bring them into connection with others (Gilligan, 1982, p. 35). Leveraging the school's Big Sister - Little sister mentorship program, the fourth-grade builders were asked to make toys for their first-grade clients. Throughout the year, builders worked closely with their clients to iterate on their toy designs, developing a close relationship and personalizing constructed toys to align with the clients' requests. The interview data indicate that builders were constantly thinking about their clients' needs and that they were proud of seeing their client's satisfaction with the handmade toys. Additionally, we have found that encouragement and emotional support from peers and teachers are also vital for young female builders in completing their projects.

Keywords
making; girls; constructionism; maker movement; diversity; elementary students

Constructionism in Different Cultures: the Case of Brazil

José Armando Valente, *jvalente@unicamp.br*
State University of Campinas, UNICAMP, Brazil

Paulo Blikstein, *paulob@stanford.edu*
Stanford University, USA

Abstract
This paper starts with the conjecture that constructionism proposed by Papert is not used in the same way in different countries and cultures. Based on Piaget's ideas it is possible that a concept might be assimilated and accommodated in terms of mental structures that were constructed based on the experiences researchers have by living in their respective cultural contexts. Thus, it is unlikely that the constructionist ideas proposed by Papert are used in the same way as in the USA, Brazil or Thailand. In Brazil, the educational context has been heavily influenced by the ideas of Paulo Freire, considered the Patron of Brazilian Education. We have several examples to show that constructionism is Brazil was influenced by the ideas of Freire, as well other factors which are particularly related to the educational system and the country social and economic situation. Initially we discuss the most important Freire's ideas that have contributed to the constitution of a new theoretical basis that support projects and activities related to the use of new technologies in education. Then we present examples of projects which was based on the constructionism ideas, although they have assumed some twists influenced by Freire's ideas. This was very explicit when one of the authors participated in a project developed by the city of São Paulo school system in which the pedagogical recommendation proposed by Freire's group of educators was to relate the theme of the project the students were developing to their cultural identity or culture context. Based on this idea it was created the contextualized constructionism concept which was used by researchers from the Nucleus of Informatics Applied to Education (NIED), State University of Campinas (UNICAMP) to support several projects such a course to prepare teachers from special education to be able to use ICT in their classroom activities; development of various software for modeling and simulating manufacturing systems to be used with factory works; to conduct studies and surveys on the use of ICT applied to the learning processes with adults who was considered as literally and technologically excluded; to study the implementation of an inquiry based learning approach, so that teachers and students could be engaged in "doing science", and using features of the laptop in a 1-1 situation.

A. Testing dye fixation on skin

B.The Tuiuiú bird

C. Chrysanthemums plant studied

D. Garbage decomposition

E. Growth of beans

Students' projects

Keywords
contextualized knowledge, Brazilian education, Paulo Freire

The Construction of Knowledge in Maker Education: A Constructivist Perspective

José Armando Valente, *jvalente@unicamp.br*
State University of Campinas, UNICAMP, Brazil

Paulo Blikstein, *paulob@stanford.edu*
Stanford University, USA

Abstract
Makerspaces are being introduced in K-12 education as an alternative to traditional approaches so that students can have more agency, engage in project-based learning activities, and generally be more active. However, "learning by doing" or by "making things" is not new. The current interest in the dissemination and popularization of makerspaces can be attributed to several trends, including the importance of educational contexts that allow for the development of STEM skills, and the fact that tools for making are becoming more accessible and easy for students to use. In makerspaces created with an educational purpose, students can learn to produce artifacts by using traditional objects and materials combined with digital and fabrication technologies. These activities are rooted in constructionist ideas proposed by Seymour Papert, which emphasize that learning is supported by the construction of artifacts that can be shown, discussed, examined, probed, and admired. The objective of this paper is to reflect on the contributions of makerspaces to the process of knowledge construction. Initially we discuss aspects related to constructionist concepts and subsequently, using Piaget's notions of conceptualization, we discuss how knowledge can be constructed in a constructionist-based makerspace. By analyzing the activities developed in an educational makerspace using digital and fabrication technologies, as shown in the figure below, one can observe that students are producing very sophisticated objects, and that it is very difficult to make these objects using traditional technologies. Furthermore, the development of these artifacts allows for the possibility of working with concepts from different knowledge areas, such as mathematics, science, and engineering. In addition to the final product, makerspaces provide the opportunity to represent the actions that are provided to the machines during the "making" process, which are expressed as concepts and strategies employed for the product to be developed. These representations constitute a "window into the mind" of the learner, allowing one to understand and to identify the knowledge that was used and, with that, help the learner reach a new level of knowledge construction.

Examples of students' projects (left to right): a robot-enacted theater play, a custom-made guitar, a "daVinci machine," and a microscope.

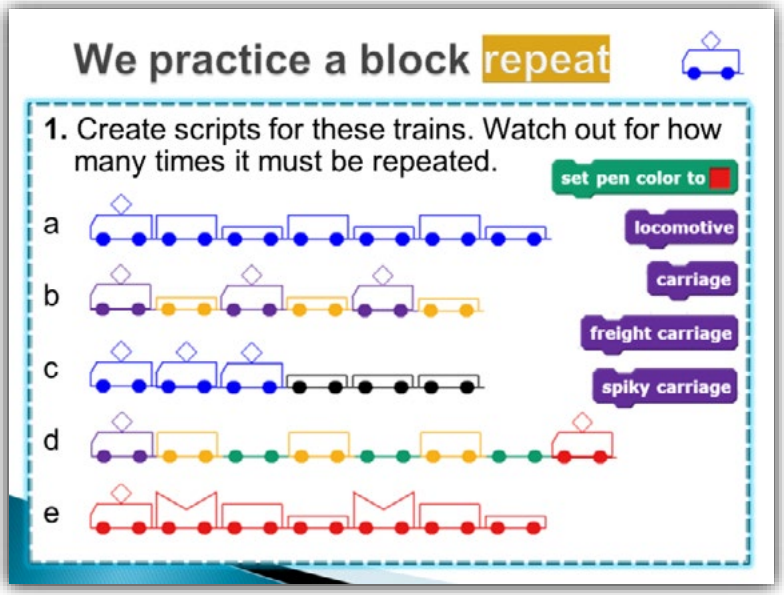
Keywords
makerspaces, maker education, fabrication technologies, knowledge construction, constructionism

Concept-building Oriented Programming Education

Jiří Vaníček, *vanicek@pf.jcu.cz*
University of South Bohemia, Czech Republic

Abstract
In this methodological paper we present a possible approach to teaching programming to 12 to 14-year-old pupils. The effort to target programming education at concept building by creating pupils' mental models has its origin in didactics of mathematics. Didactical environments are constructed that present pupils with a set of similar tasks of increasing difficulty with the same underlying concept. The world of Scratch is simplified in these environments to let pupils to focus on the given problem. Learning thus becomes more effective. An active pupil solves sets of problems in one didactical environment, is guided through various situations in which the concept is at play and builds a generic model of the concept.

A set of teacher and pupil's materials were developed within the project PRIM, whose goal is to move the compulsory school subject Informatics towards computational thinking. These materials, framed by the particular context of computing in the Czech educational system, support teaching of programming and are currently tested in schools. The paper presents concretized principles for creation of educational materials for teaching programming. These principles are illustrated in different didactical environments, on various types of pupils' activities, tasks and problems. Also experience from a pilot research study of teaching conducted with experienced teachers is presented.



This didactical environment Train shows a set of etudes for beginners to understand how to insert blocks into a loop. Each block draws one carriage. The solutions require more and more complex structure of blocks in a loop. Tasks constructed in this manner give pupils more points of view of the given concept and experience allowing them to build a generic model.

Keywords
programming education, lower secondary, concept-building, Scratch

PRACTICE PAPERS

Constructive and Collaborative Digital Storytelling for Enhancing Creativity and Cooperation In and Out of School

Francesca Agatolio, *francesca.agatolio@phd.unipd.it*
University of Padua, Dpt. of Psychology, Italy

Alfredo Asiain, *alfredo.asiain@unavarra.es*
Public University of Navarra, Dpt. of Philology & Language didactics, Spain

Alfredo Pina, *pina@unavarra.es*
Public University of Navarra, Dpt. of Math & Computer Engineering, Spain

Gabriel Rubio, *gabrielmaria.rubio@unavarra.es*
Public University of Navarra, Dpt. of Philology & Language didactics, Spain

Michele Moro, *michele.moro@dei.unipd.it*
University of Padua, Dpt. of Information Engineering, Italy

Abstract
An effective integration of technologies in the didactical context means much more than providing students with an easy access to computers. Technological tools can be used to provide students a personalised and active learning environment in accordance with the constructionist principles but, for this purpose, teachers should be aware of which type of knowledge and skill they aim to transmit and which technologies can help with that. In this paper we describe two digital storytelling experiences, held respectively in Spain and in Italy using Scratch, which provide an example of how teachers can offer an engaging learning experience without much effort. In both cases, Scratch is seen as a catalyst for different knowledge, skills and disciplines.

Two digital storytelling experiences using Scratch

The activity made in Spain, in particular, focused on the multidisciplinary aspect by the creation of stories related to the intangible cultural heritage. The Italian experience, on the other hand, was aimed at supporting creative and collaborative learning for gifted children.

Keywords
Constructionism, Digital Storytelling, Personalised Learning, Multidisciplinary Learning, Gifted Children

Towards a Framework for Educational Robotics

Julian M. Angel-Fernandez, *Angel-Fernandez@acin.tuwien.ac.at*
Vienna University of Technology, Vienna, Austria

Nikoleta Yiannoutsou, *nyiannoutsou@ppp.uoa.gr*
University of Athens, Athens, Greece

Chronis Kynigos, *kynigos@ppp.uoa.gr*
University of Athens, Athens, Greece

Carina Girvan, *girvanc@cardiff.ac.uk*
Cardiff University, Cardiff, Wales, UK

Markus Vincze, *Vincze@acin.tuwien.ac.at*
Vienna University Technology, Vienna, Austria

Abstract
Educational robotics has been considered as a field with a good potential to teach difficult concepts (e.g. friction) in appealing way. As a consequence, the interest in educational robotics has grown in the last decade, which is reflected in increasing number of robotic platforms, kits, and programming interfaces now available. Nevertheless, researches still fail on describe activities that could be used by teachers and other people with no technological fluency, who are scared by the overwhelm amount of information that made them avoid the use of robotics to teach. Moreover, most of the activities developed until now do not consider pedagogical methodologies to inform the design and implementation of them. As a direct consequence of the misinformation about the correct use of pedagogical methodologies and robotics' multidisciplinary, the number of people who master the use of robotics in education is still scant. This paper presents ongoing work on the development of a framework in the European project Educational Robotics for Science, Technology, Engineer, and Mathematics (ER4STEM). The framework aims to make evident the connection between 21st century skills, robotics and pedagogical methodologies to support the creation of pedagogical activities, which is defined in ER4STEM as an activity that has clear learning outcomes and evidence of learning, use of one or more pedagogic methodologies during the activity, and detail description of the activity. This is achieve through the critical use of tools and examples of activities developed ER4STEM.

Keywords
Educational Robotics, Framework for Educational Robotics, Pedagogical Activities, Educational Activities, Educational Robotics for STEM, and Constructionism

Think, Create and Program: Evolving to a K-9 Nationwide Computational Thinking Curriculum in Costa Rica

Carol Angulo, *carol.angulo@fod.ac.cr*
Alberto J. Cañas, *acanas@gmail.com*
Ana Gabriela Castro, *ana.castro@fod.ac.cr*
Leda Muñoz, *leda.munoz@fod.ac.cr*
Natalia Zamora, *natalia.zamora@fod.ac.cr*
Omar Dengo Foundation, Costa Rica

Abstract
As early as 1988, the National Program of Educational Informatics (PRONIE MEP-FOD) led by the Omar Dengo Foundation in partnership with the Ministry of Public Education of Costa Rica, has implemented Papert´s constructionism ideas and computer programming as part of the public schools curriculum, as a means for students to learn through constructing. This large scale program currently benefits 87.6% of the country's K-9 students, who receive two weekly lessons.

In an effort to update and revise the Program to respond to the challenges brought about by the Fourth Industrial Revolution, and to stimulate the development of the skills our students need in order to fulfill future jobs that we cannot imagine today, we are enhancing the Program into a richer curriculum that includes the understanding of computational thinking where computer programming is a fundamental methodology to achieve and exercise higher-order thinking skills, while at the same time students learn and comprehend key concepts in 'computing', all within a constructionist learning environment. Computational thinking is addressed from a broad but also deep perspective, based on the definition of key concepts, skills and attitudes from preschool to ninth grade. After the definition of the learning outcomes and the designing processes, we have been piloting this new curriculum in 107 schools and have started extending it to 1300 in 2018.

Sixth grade student using Arduino

In this paper we present advances, including methodological and conceptual foundations of the updated educational proposal and the implementation of pilot studies.

Keywords
computational thinking, programming skills, computing concepts, educational informatics, competences, problem solving, higher order thinking skills, K-9 Curriculum

Computational Thinking and Music Learning

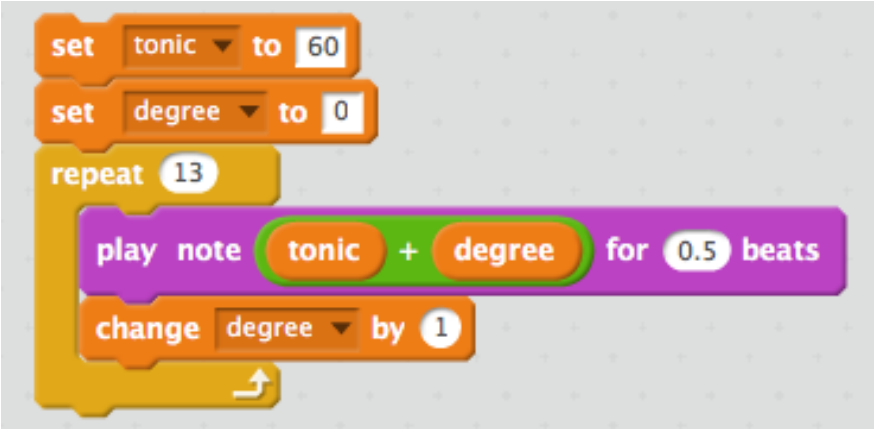
Judith Bell, *jbell@chisnallwood.school.nz*
Chisnallwood Intermediate School, Christchurch, New Zealand

Tim Bell, *tim.bell@canterbury.ac.nz*
University of Canterbury, New Zealand

Abstract
Computational thinking (CT) can be integrated with subjects outside computer science, and music is no exception. CT ideas such as decomposition, patterns, abstraction and algorithms can all be exercised in a meaningful way while at the same time engaging students with key concepts from music. This paper explores connections between computational thinking and music, and gives vignettes of creative ways to connect the two subjects in a manner that explores each subject in an authentic way.

The first example is sorting musical values (such as pitches and note lengths), which explores the ideas around sorting while exercising music reading and aural skills. We then explore the idea of giving commands to robots to position them on a giant stave to represent music; and data representation is explored by coding binary values using high and low pitches. Finally, students write computer programs to play scales and tunes, which forces them to think about both the rules around these musical concepts, as well has having to exercise programming discipline to make it sound correct and be adaptable to play in different keys.

These ideas provide students with ways to genuinely engage with both computational thinking *and* music. As well as the practical benefit of teaching two topics at once, it shows students how school subjects don't exist in isolation, and how there are aspects of thinking in common between the subjects.



A Scratch program to play a one-octave chromatic scale

Keywords
computational thinking, music theory, curriculum integration

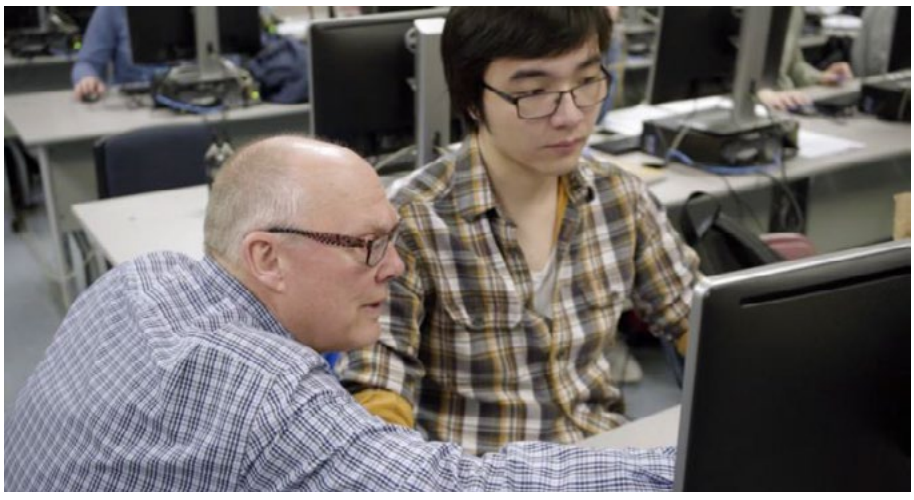
Teaching in a Sustained Post-Secondary Constructionist Implementation of Computational Thinking for Mathematics

Chantal Buteau, *cbuteau@brocku.ca*
Brock University, Canada

Ana Isabel Sacristán, *asacrist@cinvestav.mx*
Cinvestav, Mexico

Eric Muller, *emuller@brocku.ca*
Brock University, Canada

Abstract
In this practice report, we reflect and discuss the roles of, and demands on university instructors in three undergraduate mathematics computer-based courses implemented since 2001 at Brock University, Canada. These are the *Mathematics Integrated with Computers and Applications (MICA) I, II, III* courses, in which instructors create an environment that supports students' constructionist learning experiences as they design, program, and use interactive environments (i.e., microworlds) to learn and do mathematics. Using Ruthven's (2009) model on the professional adaptation of classroom practice with technology, we feature constructionist characteristics of the course design highlighting the shift from traditional, instructionist pedagogy towards one of empowering students. Since there seem to be relatively few sustained implementations of microworlds in mathematics instruction (Healy & Kynigos, 2010), this report, grounded on a continuous practice of over 15 years, contributes to our understanding of roles and demands of "ordinary" instructors in the "real" classroom, who have aimed at creating an environment for supporting students' constructionist learning experiences. In particular, this report highlights the instructor's demanding role in these student-centred courses, more so since students select their own topics for their last project in lieu of final exam, thereby having the opportunity of it being meaningful to them.



Professor Bill Ralph helping an undergraduate student engage with an authentic programming-based mathematics task during a weekly lab session.

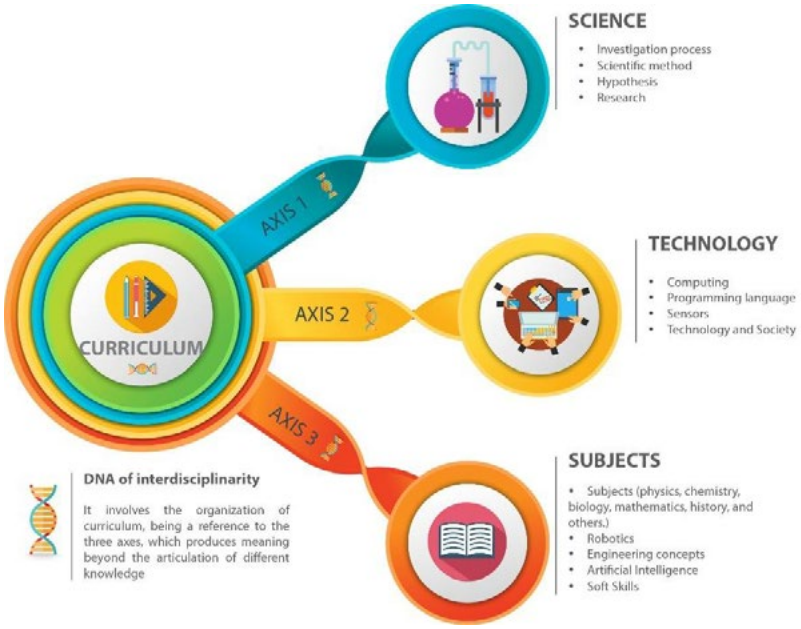
Keywords
computational thinking; constructionism; mathematics; programming; microworlds; university; teachers

Design Curriculum for Educational Robotics: Constructionist Pedagogical Experience in Formal Education

Flavio Campos, *flavio.rcampos@sp.senac.br*
University of Sao Paulo – USP, Brazil

Abstract

This paper presents a pedagogical constructionist experience in designing curriculum for educational robotics for k-12 formal education. The discussion is based on 15 years' experience of a school in Brazil, detailing a timeline of an educational robotics curriculum project development. Beginning as an afterschool program back in 2003, running as a technological resource used by math and science teachers and later integrated into curriculum from kindergarten through high school. The analyses emphasize the historical scale of a constructionist curriculum project with robotics, describing the aspects of each step and the challenges faced by the school, teachers and the administration. Therefore, the paper intends to address a curriculum design approach that drives educational robotics activities, highlighting the constructionism perspective. Pedagogy and its dimensions such as methodology, teaching and learning related to educational robotics are presented within theoretical and educational foundations. Beyond the use of robotics, deepens the discussion of curriculum design, engineering design process during robotics activities and how the integration of robotics education can boost constructionist approach in formal education.



Educational Robotics Curriculum model

Keywords

design curriculum; educational robotics; constructionism; engineering design process

Forming Concepts for Programming Conditional Statements in the Primary School

Miroslava Černochová, *miroslava.cernochova@pedf.cuni.cz*
Radek Čuma, *radek.cuma@seznam.cz*
Hasan Selcuk, *hasan.selcuk@pedf.cuni.cz*
Faculty of Education, Charles University, Prague, Czech Republic

Abstract
The Ministry of Education, Youth and Sport of the Czech Republic is currently preparing an update of curriculum documents for primary, lower and upper secondary school education in which two major changes to be made: (1) A concept of **digital literacy** will be incorporated into all school subject across the curriculum in accordance with DigComp 2.0 defined by JRC EC; (2) Instead of the existing compulsory subject ICT, **a new compulsory subject of Informatics** focused on computational thinking development, will be introduced in all levels of education. In the context of the forthcoming curricular changes, not only educational activities, but also research on the way in which pupils of different ages acquire basic information concepts has great importance.

The authors conducted a case study focussed on discovering how pupils of primary school (especially Year 3, 4 and 5) acquire, use and understand some programming conditional statements and loops (IF-THEN, IF-THEN-ELSE; REPEAT/ REPEAT-UNTIL). Programming conditional statements are undoubtedly one of the fundamental algorithmic concepts that pupils will need to understand and use in programming and algorithmic thinking development. How can pupils apply them in programming? Does it make sense to introduce these programming conditional statements into programming activities in primary education?

The research was carried out in 2017/18 at a small village school among 31 pupils (17 girls and 14 boys) of Year 3, 4 and Year 5 during 16 lessons of a compulsory subject "Work with a computer". Pupils usually worked in groups of three or four. The activities were designed in accordance with a proposal of requirements for algorithm skills and programming development in primary school education. The research was organised into four phases: (i) Preparatory phase (out of school), (ii) CSunplugged activities with a special set of paper cards and LEGO toys, (iii) Activities in a virtual environment Code.org, and (iv) Testing acquired skills and knowledge.

Findings showed that firstly, primary school pupils are able to use programming conditional statements and loops, nevertheless the Year 3 pupils can lose motivation and willigness to work if something is wrong or if they are not successful, and also they can have some linguistic barriers how to describe more details verbally their algorithmic schemes. Secondly, it has a sense to introduce these conditional statements and loops into primary education if we create for pupils conditions to link their concrete ideas based on manual operations (such as with a Lego toy) to their experinces gained in a virtual programming environment (such as Code.org).

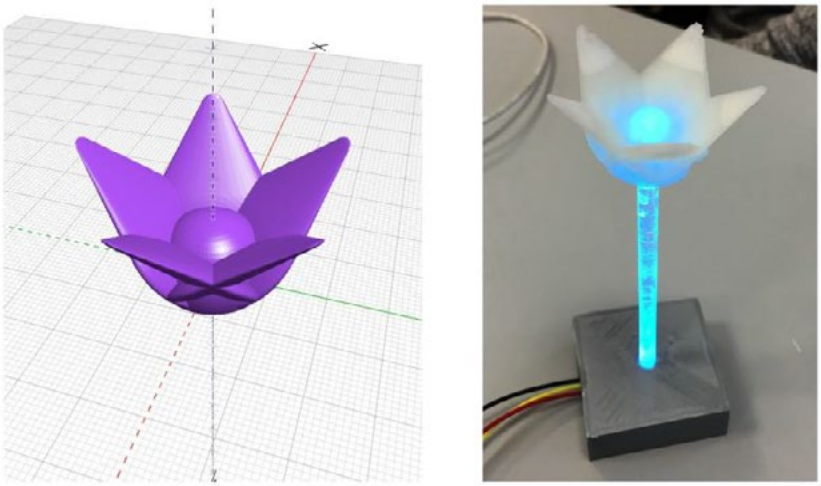
Keywords
conditional statements, loops, algorithm, programming, computational thinking, unplugged activities, Code.org

Designing Constructionist Learning Environments with Computational Design and Digital Fabrication

Christos Chytas, *christos.chytas@uni-oldenburg.de*
Computing Education Research Group, University of Oldenburg, Germany

Ira Diethelm, *ira.diethelm@uni-oldenburg.de*
Computing Education Research Group, University of Oldenburg, Germany

Abstract
Makerspaces like fab labs (digital fabrication laboratories) are open workshops that promise to democratize the means of production and technical knowledge. Even though such laboratories are often seen as innovation spaces for small business they are receiving increasing attention as informal learning environments for STEAM (Science, Technology, Engineering, Art, Mathematics) subjects. "Making" as a set of learning activities roots in learning theories of educators like Seymour Papert. Inspired by Papert's constructionist learning theory, we designed, implemented and evaluated workshops on digital fabrication and computational design for children and youth. The workshops' goal was to introduce computational concepts and programming as means of personal expression through the creation of computational design models that could be fabricated in our labs.



Generated 3D parametric model of a flower which was 3D printed to make an interactive artistic project

The workshops' concept provided opportunities for creativity, personal expression, collaboration and content rich learning activities to create artistic, practical or entertaining artifacts. We identify the elements of maker tools and culture that enhanced the learning experience in our workshops. We discuss implications and challenges of these elements for educators who wish to use digital fabrication for programming learning activities.

Keywords
Constructionism; Computational Design; Digital fabrication; The maker movement; Computing Education; Fab Lab

Developing Mathetic Content Knowledge using an Emergent Systems Microworld

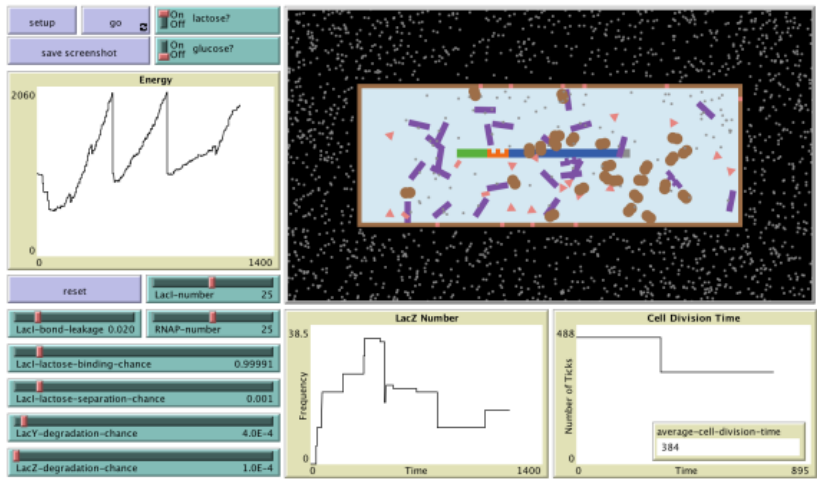
Sugat Dabholkar, *sugat@u.northwestern.edu*
Learning Sciences and Center for Connected Learning, Northwestern University, USA

Gabriella Anton, *gabby.anton@gmail.com*
Learning Sciences and Center for Connected Learning, Northwestern University, USA

Uri Wilensky, *uri@northwestern.edu*
Learning Sciences, Computer Science and Center for Connected Learning,
Northwestern University, USA

Abstract

In this paper, we define and develop a theoretical construct Mathetic Content Knowledge (MCK) in the context of children's learning. We discuss acquiring MCK using a special type of constructionist learning environment, which we call Emergent Systems Microworlds (ESMs). ESMs allow students to engage with emergent phenomena in an exploratory way. We argue that the students who participated in our ESM-based curricular unit, GenEvo about Genetics and Evolution not only learned disciplinary core ideas, but also possibly developed mathetic insights into how to learn by engaging in the scientific inquiry process. The GenEvo curriculum incorporates a series of computational models designed using NetLogo that follow the agent-based modeling approach to emergent systems. In this curriculum, students design and conduct computational experiments in the ESM learning environment to figure out the answers to the guiding questions collectively build towards the ideas about emergent properties in the ESM.



A screenshot of NetLogo Genetic Switch Model (Dabholkar et. al, 2016), showing a representation of a bacterial cell with parts of the genetic circuit - DNA and proteins

We argue for the importance of fostering Mathetic Content Knowledge, knowledge of how to learn by engaging in discipline specific inquiry practices, in general, and specifically for science education. We also discuss how systematic exploration of computational models using ESM learning environments could be an effective way to develop science MCK.

Keywords

Mathetic Content Knowledge; Emergent Systems Microworlds, Science inquiry practices; Constructionism; Design

Making Together: Cultivating Community of Practice in an All-Girl Constructionist Learning Environment

Caitlin Davey, *csd2126@tc.columbia.edu*
Teachers College, Columbia University USA

Sawaros Thanapornsangsuth, *st2839@tc.columbia.edu*
Teachers College, Columbia University USA

Nathan Holbert, *holbert@tc.columbia.edu*
Teachers College, Columbia University USA

Abstract
This practice paper investigates instructional practices that support relationships among young female makers. It summarizes a design-based research study where a group of all-female makers constructed toys for younger students in their immediate school community. Makers supported one another through contributing ideas, building together, sharing expertise, and providing helpful encouragement. The finding shows that the classroom practices extended beyond physical construction to include playful interactions amongst makers such as: switching projects, working together outside class time, and a myriad of other activities both related and unrelated to making. Through the process of making, they developed individual expertise that contributed to the sharing of knowledge within their classroom community. Drawing on literature from constructionist design paradigms and community of practices, data from this Making and Engineering class describes an emergent community of makers. Additionally, this paper highlights the value of makers creating personally and socially meaningful projects in collaboration with others. Finally, we describe the flexible and playful environment of the Making and Engineering classroom that contributed to how makers shape their shared practices.

Keywords
Constructionism; Community of practice; maker education; girls in making

Assessment of Modeling Projects in Informatics Class

Natasa Grgurina, *n.grgurina@rug.nl*
University of Groningen, Netherlands

Erik Barendsen, *e.barendsen@cs.ru.nl*
Radboud and Open University, Netherlands

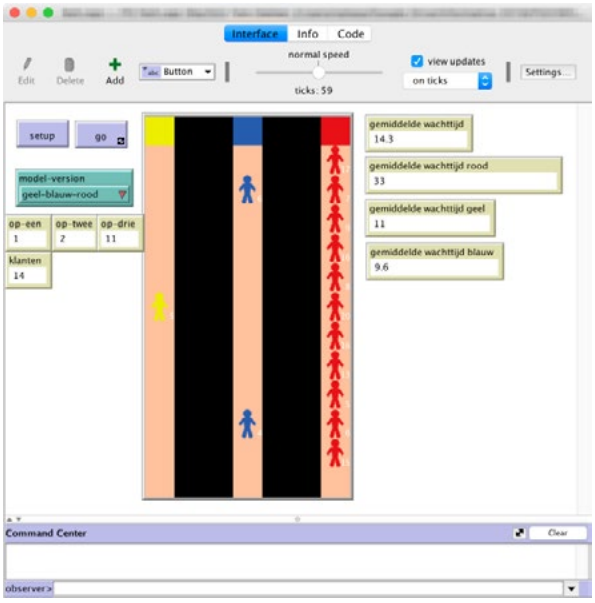
Cor Suhre, *c.j.m.suhre@rug.nl*
University of Groningen, Netherlands

Klaas van Veen, *klaas.van.veen@rug.nl*
University of Groningen, Netherlands

Bert Zwaneveld, *zwane013@planet.nl*
Open University, Netherlands

Abstract

The introduction of the new Informatics curriculum in the Netherlands in 2019 raises the need for new teaching material that includes practical assignments and guidelines for their assessment. As a part of our research project on teaching Computational Science (modeling and simulation), we participate in these efforts and developed a curriculum intervention and an assessment instrument consisting of a practical assignment and grading rubrics to assess student's level of understanding. The rubrics we developed can be used both for formative and summative assessment. In this paper we describe the design of this assessment instrument and indicate further research directions focusing on validation of this instrument.



Student's model in NetLogo for bank counter problem

Keywords

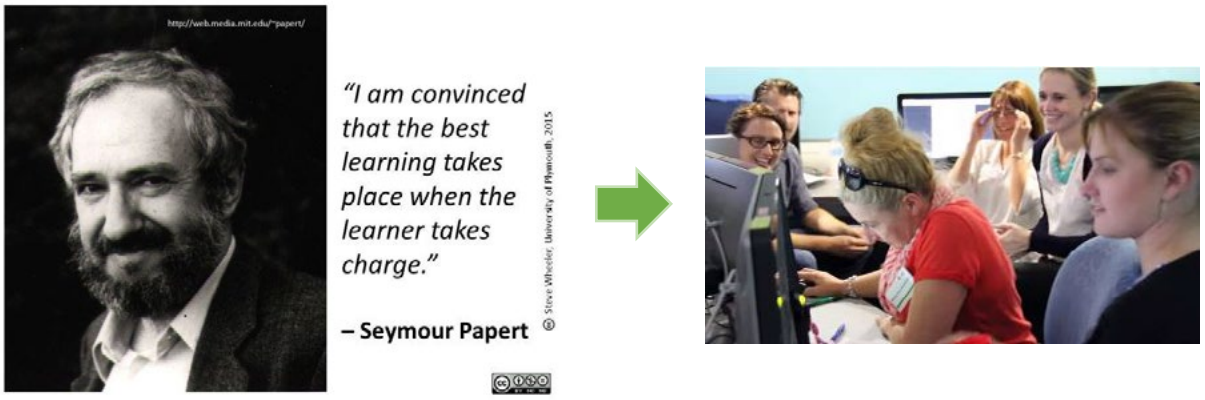
Modeling and simulation, NetLogo, Assessment, SOLO-taxonomy

Constructionist Experiences in Teacher Professional Development: A Tale of Five Years

Daniel Hickmott, *daniel.hickmott@uon.edu.au*
School of Education, The University of Newcastle, Australia

Elena Prieto-Rodriguez, *elena.prieto@newcastle.edu.au*
School of Education, The University of Newcastle, Australia

Abstract
Computational thinking and coding have recently become compulsory elements in the Australian K-8 curriculum that should be taught using 'authentic learning challenges' (ACARA, 2018a). However, very few teachers, particularly in the primary school setting, have been schooled on computational thinking or coding and rarely possess pedagogies to teach them authentically. A range of professional development opportunities are currently being offered to impart this knowledge, both for content and pedagogy. In this paper, we provide an account of the evolution we have experienced when designing and improving professional development workshops for teachers in coding and computational thinking. We reflect on our challenges and successes, and attest that it was only after 'discovering' Constructionism in late 2015 that we have been able to prepare activities that truly emulate the authentic learning experiences that teachers are required to use in their classrooms.



Learning from pioneers: the influence of Seymour Papert on our professional development

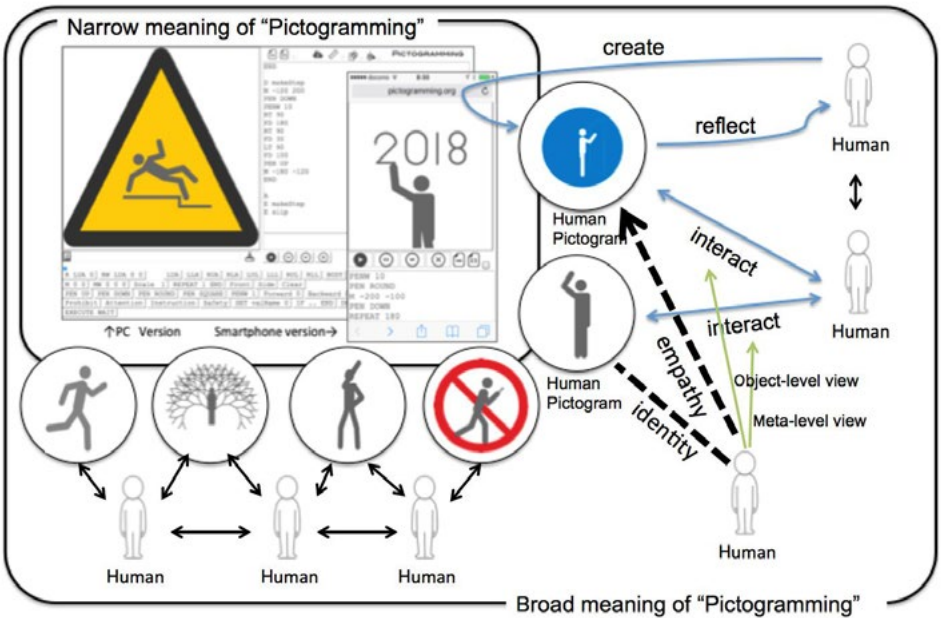
Keywords
Teacher professional development; Constructionism; computing; programming

Pictogramming: Learning Environment Using Human Pictograms Based on Constructionism

Kazunari Ito, *kaz@si.aoyama.ac.jp*
Aoyama Gakuin University, Japan

Abstract

We have been developing a pictogram authoring tool "Pictogramming." Pictogramming is based on two words: "pictogram" and "programming." The basis of this application is the use of a human-shaped pictogram (i.e., a "human pictogram"). Pictograms are an abstract representation of a person or an object. Creating a posture of a human pictogram is strongly associated with the embodies knowledge. And the works with free themes are created with one's ego syntonic and are firmly and positively based on ones' culture. A useful pictogram is made by strongly connecting with one's culture. That is, constructing human pictograms are very familiar with syntonic learning insisted on by Papert. We can expand this concept beyond the limits of the proposed application (we call this the "narrow meaning of Pictogramming") to any activities that correspond to human pictograms, which is termed the "broad meaning of Pictogramming."



Narrow and broad meaning of Pictogramming

In this paper, we first explain the function of the application and discuss the broad meaning of Pictogramming from the viewpoint of constructionism.

Keywords

Logo, Human pictogram, Syntonic learning, Duplication of Viewpoint, Educational environment

Human Pictogram Unplugged: Unified Learning Environment of Computer Science Unplugged Using Human Pictograms

Kazunari Ito, *kaz@si.aoyama.ac.jp*
Aoyama Gakuin University, Japan

Aoi Yoshida, *aoi@si.aoyama.ac.jp*
Aoyama Gakuin University, Japan

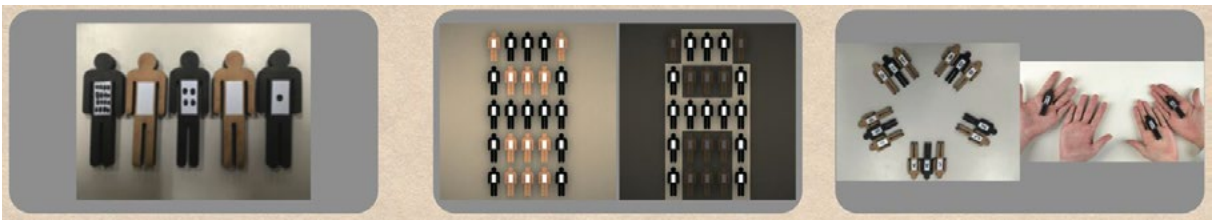
Takashi Yoneda, *yoneda@port.kobe-u.ac.jp*
Kobe University Secondary School, Japan

Yuichi Oie, *c8116002@aoyama.jp*
Aoyama Gakuin University, Japan

Abstract

We have been developing a unified learning environment for computer science unplugged (CSU) using human pictograms called “Human Pictogram Unplugged” (HPU). CSU is a methodology of teaching computer science without the use of personal computers. One of HPU’s characteristics allows students to conduct activities with a set of unified human-shaped pictograms (i.e., a “human pictogram”). Educators usually must prepare and maintain teaching materials for each CSU activity: a balance for “sorting” activity, worksheets for “image representation,” and special cards for “binary numbers,” for example. To decrease this economic and temporal burden is an important and notable issue in education. Human pictograms are abstract representations of a people. Activity themes using human pictograms would at once be more strongly connected to one’s experience and knowledge and also firmly and positively based on one’s culture. Therefore, HPU will promote syntonic learning.

Activity examples of Human Pictogram Unplugged



Binary Number *Image Representation* *Routing and Deadlock*

In this paper, we explain the concept of HPU, show example activities of HPU, and discuss HPU’s effects and characteristics from a constructionist viewpoint.

Keywords

Human Pictogram, Computer Science Unplugged, Syntonic learning, Duplication of Viewpoint, Educational environment

Some Reflections on Designing Constructionist Activities for Classrooms

Yasmin B. Kafai, *kafai@upenn.edu*
University of Pennsylvania, 3700 Walnut Street, Philadelphia, PA, 19104, USA

Deborah A. Fields, *deborah.fields@usu.edu*
Utah State University, 2830 Old Main Hill, Logan, UT, 84322, USA

Abstract
In this paper, we present our guiding principles for designing a constructionist curricular unit called "Stitching the Loop" with electronic textiles which introduces high school students to key concepts in crafting, circuit design and computing. Our principles were to design for (1) engagement by promoting interest-driven designs; (2) expression by putting aesthetics first; (3) depth by developing challenging content within constraints; (4) multiple experiences for providing opportunities for practice; (5) audience by sharing designs; (6) collaboration by having students help other students; (7) reflection by including design notebooks and portfolios; (8) failure by having students and teachers model and celebrate mistakes; (9) practicalities that transform classrooms into maker studios; and (10) iterations, iterations, and more iterations. Over three years, we worked together with dozens of high school teachers and hundreds of students in implementing and revising classroom activities in which students design and craft a series of individual and collaborative electronic textiles projects. Situated within the larger framework of the year-long *Exploring Computer Science* curriculum, we illustrate how these guiding principles fostered an equity-focused and inquiry-oriented pedagogy through which teachers could contribute to and support students' learning.

Gallery of sample student projects in the "Stitching the Loop" e-textiles unit

Keywords
curriculum; coding; computer science education; electronic textiles, constructionism, maker movement

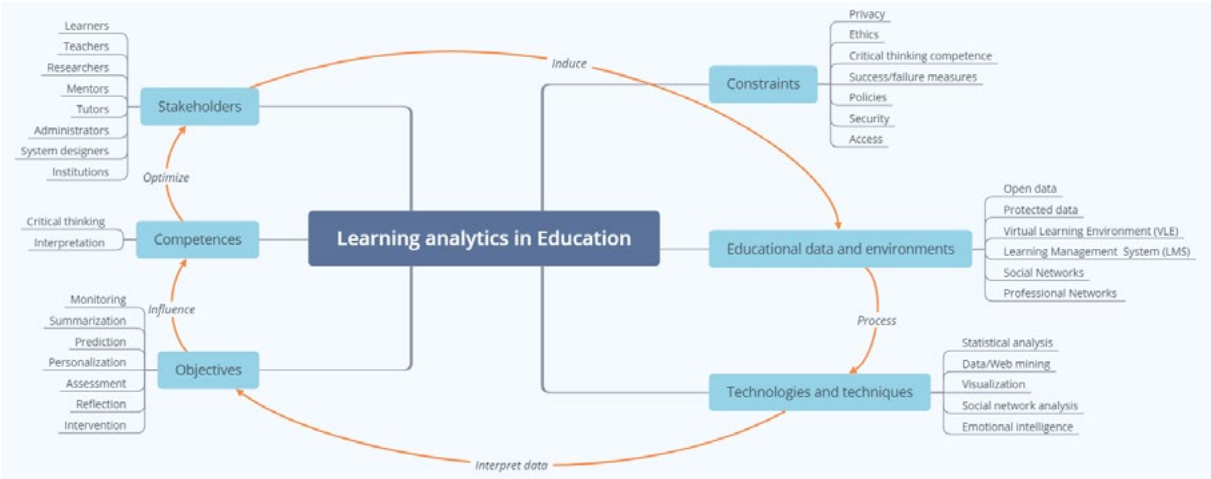
Learning Analytics in Education: Objectives, Application Possibilities and Challenges

Aleksandra Klašnja-Milićević, *akm@dmi.uns.ac.rs*
Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovica 3, Novi Sad, Serbia

Mirjana Ivanović, *mira@dmi.uns.ac.rs*
Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovica 3, Novi Sad, Serbia

Abstract

With an increased amount of educational data in all domains of human activities, Learning Analytics (LA) become a powerful mechanism for supporting learners, instructors, teachers, learning system designers and developers to better understand and predict learners' needs and performances. In this paper, we analyse the significant dimensions and objectives of LA, application possibilities and some challenges to the beneficial exploitation of educational data. We identify the required skills and capabilities that make meaningful use of LA techniques and technologies in this domain. They can act as a useful guide for setting up LA services in support of educational practice and learner guidance, in quality assurance, curriculum development, and in improving learning process effectiveness and efficiency. Furthermore, this paper proposes the most important constraints that affect LA technologies in education.



Learning Analytics in Education

Keywords

Learning Analytics (LA); Education; Application; Learning Environments; Effective Learning System

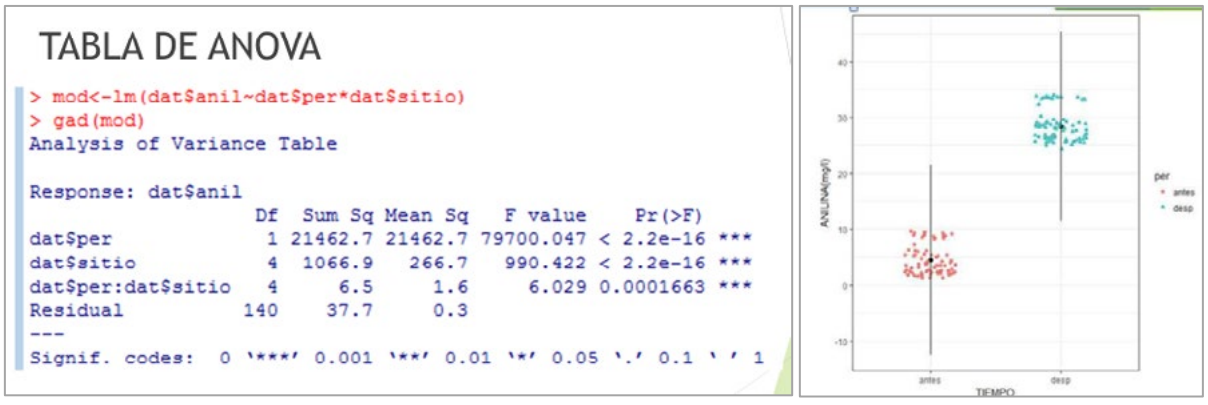
Assessing Learning through Exploratory Projects in Cnstructionist Rbased Statistics Courses for Environmental Science Students

Maite Mascaró, *mmm@ciencias.unam.mx*
UMDI-Sisal, National Autonomous University of Mexico (UNAM), Mexico

Ana Isabel Sacristán, *asacrist@cinvestav.mx*
Centre for Research and Advanced Studies (Cinvestav), Mexico

Abstract
For years we have been developing sequences of constructionist and collaborative activities where university environmental science students engage in computational programming tasks (using R code – http://www.r-project.org/) for exploring and learning statistical concepts and ideas. In this practice paper, we present a project-based method that we use for assessing students´ learning and appropriation of those statistical concepts.

The exploratory projects that we use for assessment, are open-ended realistic problems in environmental contexts. In these projects, students, working collaboratively in teams of up to 4 members, are expected to make choices of the most appropriate statistical procedures and graphic representations to be used in portraying relevant statistical features in the data, and construct statistical models using any tool they choose, such as R code. They then have to defend the line of reasoning underlying their choices, through a formal presentation using their own communication resources, and where other students question and discuss the methods, results and interpretations presented. Such projects can be considered to be thought-revealing activities; i.e. they serve as a window into students' thinking, and thus are means for assessing their learning. They also constitute an approach that is more in accordance with constructionist principles, since students explore and construct freely. Such projects thus, not only serve as a means for us to assess students' learning, but can be, in themselves, learning experiences that help reinforce what has already been learned.



Students using R programming for statistical explorations of a problem (such as constructing ANOVA tables and various graphical representations), in an open-ended project for assessment of their learning.

Keywords
statistics education, assessment, exploratory project, R-programming, constructionism

How Students Struggled with Preparation of Activities for a Leisure Time Robotic Workshop

Karolína Mayerová, *mayerova@fmph.uniba.sk*
Comenius University, FMFI, Bratislava, Slovakia

Michaela Veselovská, *veselovska@fmph.uniba.sk*
Comenius University, FMFI, Bratislava, Slovakia

Abstract
In this article we describe university compulsory elective course, which we taught in the winter semester of 2017. We briefly explain specific conditions, which encourage transformation of the organization of the course. The aim of this article is to provide an overview of the process and results of a case study on integrating authentic learning into mentioned course. Not only from point of view of usage robotic kits during course lessons but also from the point of view of creating specific activities for leisure time workshop. This workshop was taught by our student (let's call her Jane), who also conducted observations there. In our research we used qualitative methods of data collection and data analysis including observations, recorded video, students' work process and outcomes (created activities). We introduce several activities with remarks from conducted observations. These activities were created with constructionist approach by our students during the course. We present several findings from Jane, from the teacher' point of view and also from pupils' point of view. She identified several requirements for content of activities for robotic workshop, which should include solutions of all the tasks in the activities and methodological guidelines for teaching.

Examples of pupils' work on activities with different types of robotic kits within robotic workshop and examples of created programs to control them

Based on Jane observations, we found out that number of pupils in the workshop was growing especially because of attractive nature of created activities. We hope that such increased internal motivation could help pupils to acquire a positive relationship to the compulsory school subject Informatics and to direct them in career growth.

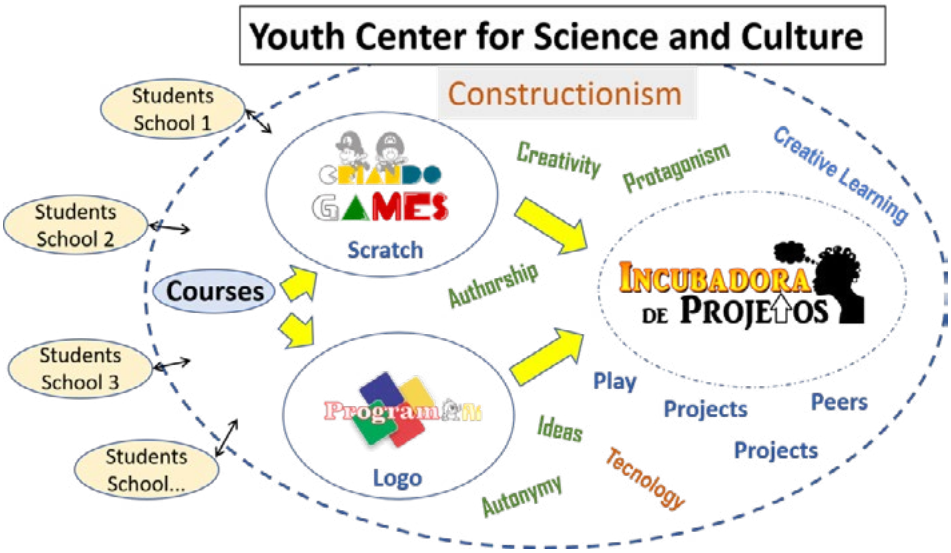
Keywords
University elective course within teacher's training; educational robotics; after school activities for primary and lower secondary school pupils; authentic learning

Constructionism as an Epistemological Option in Courses of Youth Center for Science and Culture – Bahia – Brazil

Elmara Pereira de Souza, *elmarasouza@gmail.com*
Youth Center for Science and Culture – Vitória da Conquista – Bahia, Brazil

Luísa Souza Moura, *luisasouzamoura@gmail.com*
University of São Paulo (USP) – São Paulo, Brazil

Abstract
The goal of this paper is to report on the practice of the Youth Center for Science and Culture, Vitória da Conquista, Bahia, Brazil, a public school for secondary school students. The Youth Center uses a constructionist perspective as an epistemological basis in their courses. The students are the main characters in the learning process and they use the coding languages Logo and Scratch in their projects' development.



Pedagogical structure of the courses of Youth Center for Science and Culture – Vitória da Conquista – Bahia - Brazil

This experience shows that it is possible to create spaces of authorship so that the public school's students don't turn out to be just users of the technologies, but authors of socially relevant content, that can change their lives and that can provide future perspectives.

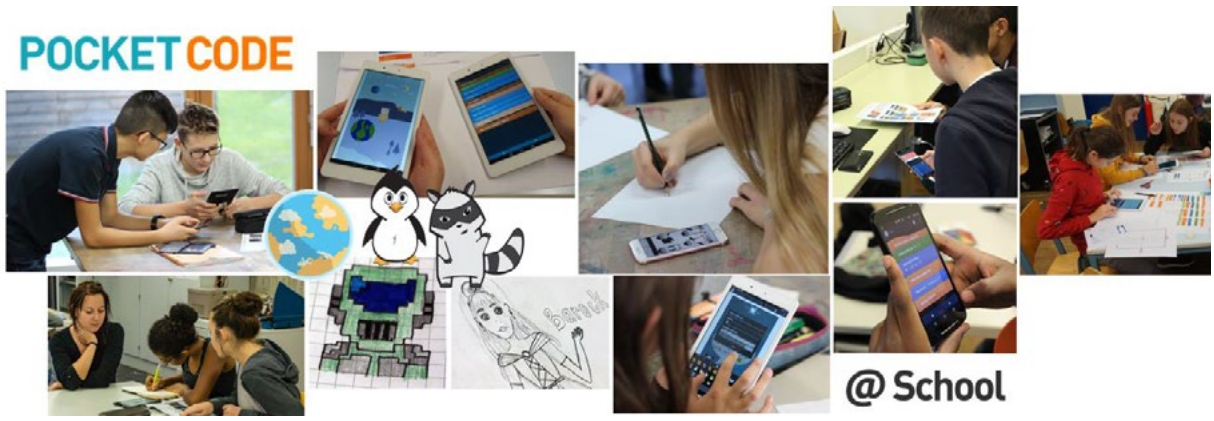
Keywords
Constructionism; authorship; protagonism of young people

Female Teenagers and Coding: Create Gender Sensitive and Creative Learning Environments

Bernadette Spieler, *bernadette.spieler@ist.tugraz.at*
Institute of Software Technology, Graz University of Technology, Austria

Wolfgang Slany, *wolfgang.slany@tugraz.at*
Institute of Software Technology, Graz University of Technology, Austria

Abstract
The number of women in technical fields is far below the average number of males, especially in developed countries. Gender differences in STEM are already present in secondary schools in students aged between 12 to 15 years. It is during this intermediate female adolescence that girls begin to make critical career choices, which therefore makes this a key age to reinforce them and reduce the gender disparities in ICT. Acquiring computational thinking (CT) skills, particularly coding, is important for building a positive economic, developmental, and innovative future. To address the gender bias in schools, one of the goals of the European H2020 project No One Left Behind (NOLB) included integrating Pocket Code, a free open source app developed by the non-profit project Catrobat, into different school lessons. Through game design, Pocket Code allows teenage girls to incorporate diversity and inclusiveness, as well as the ability to reflect their cultural identity, their likes, and their ways of interacting and thinking. To evaluate the impact of the use of the app in these courses, we captured the results on engaging girls in design and coding activities. For this paper, the authors present the data of surveys via a qualitative content analysis during the second cycle of the project. The results let the researchers conclude that the organization and the setting of the coding courses (for example, guidance and supporting material, freedom of choice) had much more influence on female students' engagement than the coding aspects or the app itself. In contrast, male students more frequently mentioned missing features in the app, and stated that they liked the coding. With a focus on female teenagers, the results allow us to conclude that a suitable classroom setting is significantly more important for them than the coding tool itself.



Providing inclusive coding environments for female teenagers

Keywords
Pocket Code, Game Design, Gender Inclusion, Coding, Mobile Learning, STEM, Social Inclusion, Constructionism, Girls, Learning Environment

Visualizing Mathematics with the MathBot: a Constructionist Activity to Explore Mathematical Concepts through Robotics

Christina Todorova, *tina@esicenter.bg*
European Software Institute – Center Eastern Europe, Sofia, Bulgaria

Carina Girvan, *girvanc@cardiff.ac.uk*
School of Social Sciences, Cardiff University, Wales, UK

Nikoleta Yiannoutsou, *nyiannoutsou@ppp.uoa.gr*
Marianthi Grizioti, *mgriziot@ppp.uoa.gr*
UoA ETL, National and Kapodistrian University of Athens, Athens, Greece

Ivaylo Gueorguiev, *ivo@esicenter.bg*
Pavel Varbanov, *pavel@esicenter.bg*
George Sharkov, *gesha@esicenter.bg*
European Software Institute – Center Eastern Europe, Sofia, Bulgaria

Abstract
In this practice paper, we aim to share our experience with the design and implementation of constructionist educational robotics activities tailored to primary school students (4th grade, age 9-11 years) implemented in a series of robotics workshops, which took place within a real school setting in Sofia, Bulgaria.

Through this contribution, we will further present an activity plan, which involves student engagement with mathematical concepts (angle measuring and properties of the circle) in order to program the behaviour of a robot. Our paper reports insights on the implementation of the activity plan focusing students' evaluation of their experience during the workshop. These insights are drawn from quantitative data from 131 participants (63 boys and 68 girls), capturing the overall student attitude.

The activity plan behind this set of educational robotics workshops was designed, adapted and piloted in alignment to the guidelines of the Bulgarian national curriculum for mathematics for the 4th grade. However, it could function as a practical example, which could be adapted, enriched and modified to benefit other age groups, nationalities and desired learning outcomes.

Keywords
Educational robotics; mathematics; programming; Scratch; primary education

Ethnomathematics in Teacher Education: Analysis and Construction of Geometric Ornaments

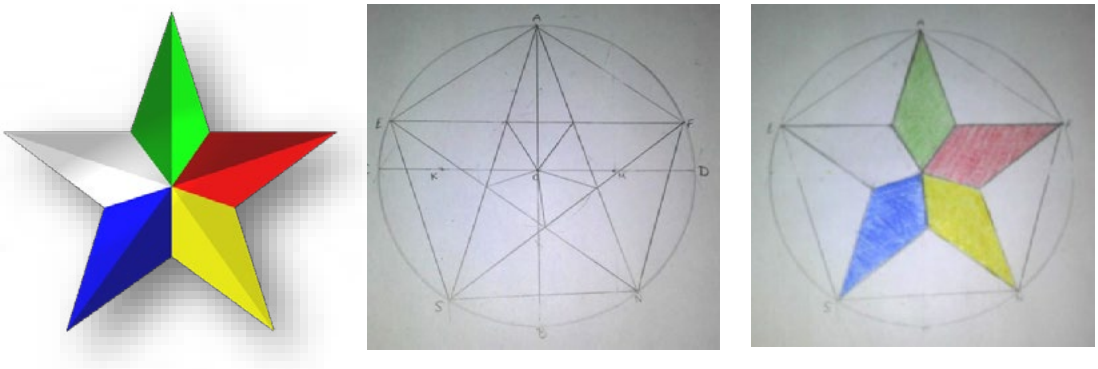
Igor Verner, *ttrigor@technion.ac.il*
Technion – Israel Institute of Technology, Israel

Khayriah Massarwe, *massarwe@technion.ac.il*
Technion – Israel Institute of Technology, The Arab Academic College for Education, Israel

Daoud Bshouty, *daoud@technion.ac.il*
Technion – Israel Institute of Technology, Israel

Abstract
This paper presents a study in which we developed, implemented, and evaluated a teacher education course for teaching geometry based on the ethnomathematics approach. The participants, prospective and in-service teachers with different cultural backgrounds studied geometry through analysis and construction of geometric ornaments from diverse cultures, and acquired knowledge and skills in multicultural education. The students inquired geometrical properties and symbolism of the ornaments from the cultures they chose. In the study we analysed motivating desires observed in the course using the theory of engagement structures proposed by Goldin and colleagues. We found that some of the engagement structures that are typical for conventional mathematics classes, emerged in our course while other structures were not observed. We propose a new, additional engagement structure to embody motivational desires arising from multicultural interactions in diverse classes.

The figure below presents the Druze Star, the prominent symbol of the Druze culture. It also shows the geometric construction of the Star performed by a Druze student. She figured out the meaning of colours of the Star: green symbolizes nature; red symbolizes courage and love; yellow symbolizes enlightenment; blue indicates patience and brotherhood; white indicates reconciliation and peace.



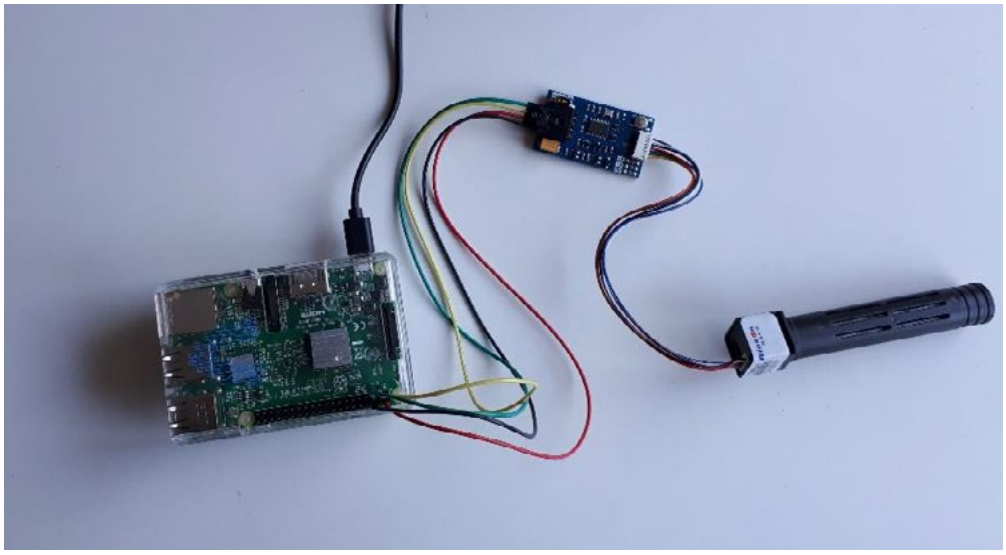
Keywords
Ethnomathematics; geometric ornaments; teacher education; learning engagement.

Coding to Learn – Informatics in Science Education

Michael Weigend, *mw@creative-informatics.de*
Holzkamp-Gesamtschule Witten, Germany

Abstract
Programming is a way to explore and elaborate scientific content. Software projects might (to some extend) replace traditional pencil-and-paper textbook problems that require algebra skills. This contribution discusses a Python workshop which has been conducted with 34 students from three German high school chemistry classes in grades 11 and 12. The participants (mostly programming novices) developed interactive programs solving quantitative Chemistry problems. They claimed to enjoy programming, the girls almost as much as the boys. A majority agreed that programming implies practising logical thinking, precise communication and creativity as well as elaborating chemistry knowledge.

Written material (tutorials) for these kinds of workshops may include interactive exercises, prototype projects with verbal and visual explanations and suggestions for more advanced projects. With a Raspberry Pi and digital sensor units (e.g. light sensor, NDIR carbon dioxide sensor), students can create systems that perform scientific experiments semi-automatically. Programming projects in science classes need to be simple. But they may still be relevant, make science education more interesting and learning more individual.



Raspberry Pi with NDIR CO_2-sensor module

Keywords
science education, programming, computational thinking

Media Parkour – Experiential Learning Activities for Media Education

Michael Weigend, *mw@creative-informatics.de*
Holzkamp-Gesamtschule Witten, Germany

Fenja Göcking, *fenja.goecking@gmx.de*
Alexander Knuth, *knuthalex@web.de*
Patrick Pais Pereira, *patrick-pais-pereira@hotmail.com*
Laura Schmidt, *laurakathleen2305@gmail.com*

Abstract
This contribution presents a sequence of quick and easy experiential learning activities for sixth graders, which were developed, performed and evaluated by a high school pedagogy class. The activities mainly take place outside in the schoolyard and challenge creativity, computational thinking and collaboration. The learning arrangements are inspired by ideas of experiential learning and cover topics from media education like appropriate reactions to cyber bullying, data transmission between mobile phones, recognizing commercial advertisements, falsification of news etc. The Media Parkour was conducted with almost fifty sixth-graders in the age of 11 to 12. They performed the activities in teams of four and then rated them in a questionnaire.

Keywords
media education; computational thinking; gamification; experiential learning

A Creative Learning Sequence in an Introductory Programming MOOC

Elisabeth Wetzinger, *elisabeth.wetzinger@tuwien.ac.at*
Vienna University of Technology, Austria

Gerald Futschek, *gerald.futschek@tuwien.ac.at*
Vienna University of Technology, Austria

Bernhard Standl, *bernhard.standl@ifs.tuwien.ac.at*
Vienna University of Technology, Austria

Abstract
Lecture-videos, tutorials, and summative examinations frequently give Massive Open Online Courses (MOOCs) the characteristics of teacher-centered instruction. The amount of individual feedback is often reduced to a minimum and most of the practical tasks include multiple-choice quizzes. This is mainly due to the big amount of anonymous and heterogeneous learners. Although a large (massive) number of participants with different goals and pre-education on one hand and compared to that a small group of advisors, on the other hand, suggest simple learning tasks, this situation is not satisfying in respect to our expectation of a constructive and creative learning environment.

In January 2017, we started to design and develop a MOOC for learning computer programming, addressing high-school students without or with little experience in programming intending to study a STEM bachelor programme at our college. Hence, our goal was to conform the heterogeneous levels of pre-knowledge in computer programming of our freshmen students. We decided to use *Processing*[5] as programming language, because it supports an easy learning start to programming by using graphics and animations and allows for an easy transition to Java-like programming languages later on.

As part of the course we designed a learning sequence that addresses the participants' creativity and includes peer feedback, and communication with other participants. We asked students already at a very early stage to create an artwork using previously presented basic graphical procedures. The "artistic" results has to be shown to the other participants combined with some personal remarks. Then they are asked to choose one of the artworks, create a copy of it by coding and provide feedback to the original artist why his or her work was chosen and what challenges they faced in creating a fake copy, optionally completed by further remarks.

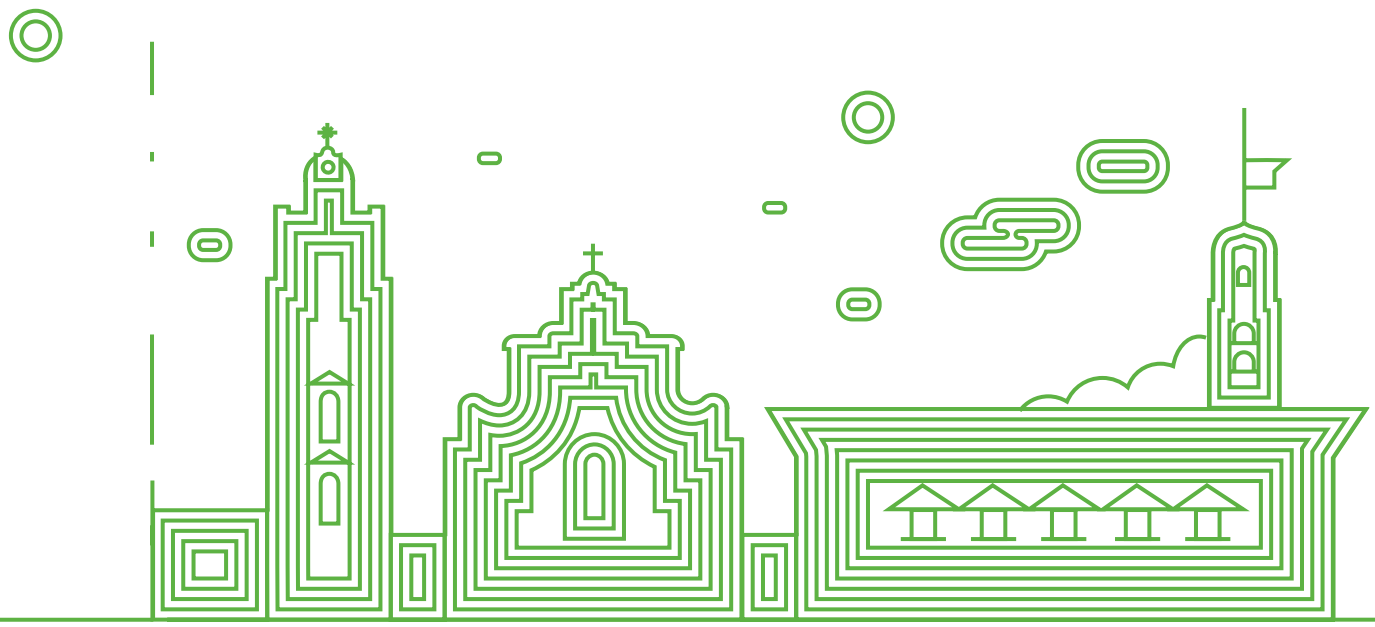
Example of an original artwork (left) and its "fake" (right)

In this paper, we present our experience implementing the exercise, results of this learning sequence and selected feedback of our MOOC students concerning the assignment.

Keywords
creative learning sequence; MOOC; learning computer programming

[5] http://processing.org

POSTERS



Educational Robotics for STEM: From Workshops to Curricula and Framework

Carina Girvan, *girvanc@cardiff.ac.uk*
School of Social Sciences, Cardiff University, Wales, UK

Wilfried Lepuschitz, *lepuschitz@pria.at*
Practical Robotics Institute Austria, Austria

Ivaylo Gueorguiev, *ivo@esicenter.bg*

Christina Todorova, *tina@esicenter.bg*
European Software Institute – Center Eastern Europe, Sofia, Bulgaria.

Chronis Kynigos, *kynigos@ppp.uoa.gr*
Marianthi Grizioti, *margrizioti@gmail.com*
UoA ETL, National and Kapodistrian University of Athens, Athens, Greece.

Angele Giuliano, *angele@acrosslimits.com*
Annalise Duca, *annalise@acrosslimits.com*
AcrossLimits, Malta

Julian M. Angel-Fernandez, *julian.angel.fernandez@tuwien.ac.at*
Markus Vincze, *vincze@acin.tuwien.ac.at*
Technical University Wein, Vienna, Austria

Abstract
This poster presents the outcomes of the first two years of a three-year project exploring and refining the use of educational robotics to engage young people in STEM education. It shows the development of activity plans which leverage constructionist concepts such as powerful ideas, objects to think with and the construction and sharing of personally meaningful artefacts in order to explore, test and extend understanding within STEM domains. Workshops in which these activity plans have been tested and refined, have so far engaged over 3,000 children between the ages of 7 and 18 in six European countries. Additionally, conferences and competitions for young people have been held each year where they compete and collaborate with others. These workshops and competitions have been systematically evaluated through QUAL+quant mixed methods, in order to inform the development of future activities and refinement of existing ones. Through this process we identify key components of successful educational robotics activities for STEM education. These have, in-turn, informed the design of a framework for educational robotics. The activity plans have informed the bottom-up development of a generic curriculum for educational robotics in STEM and are available on a dedicated repository. This poster presents a snapshot of each of these project outcomes.

Keywords
Constructionism, educational robotics, science, technology, engineering, mathematics, learning, STEM

Towards a Generic Curriculum for Educational Robotics in STEM: From Scientific Concepts to Technologies and Powerful Ideas

Ivaylo Gueorguiev, *ivo@esicenter.bg*
Christina Todorova, *tina@esicenter.bg*
European Software Institute – Center Eastern Europe, Sofia, Bulgaria

Nikoleta Yiannoutsou, *nyiannoutsou@ppp.uoa.gr*
Xristina Greka *xristgreka@gmail.com*
UoA ETL, National and Kapodistrian University of Athens, Athens, Greece

Pavel Varbanov, *pavel@esicenter.bg*
George Sharkov, *gesha@esicenter.bg*
European Software Institute – Center Eastern Europe, Sofia, Bulgaria

Carina Girvan, *girvanc@cardiff.ac.uk*
School of Social Sciences, Cardiff University, Wales, UK

Julian M. Angel-Fernandez, *julian.angel.fernandez@tuwien.ac.at*
Vienna University of Technology, Vienna, Austria

Lisa Vittori, *vittori@pria.at*
Practical Robotics Institute Austria, Austria

Annalise Duca, *annalise@acrosslimits.com*
AcrossLimits, Malta

Abstract
This paper and its corresponding poster present a "work in progress" concept for the visualization of 19 activity plans, into a generic curriculum map for teaching STEM concepts through constructionist robotics activities. There are six educational paths that represent potential use cases and these have been validated through 148 educational robotics workshops implemented with children between the ages of 7 and 18 in six European countries.

Keywords
educational robotics, curricula, robotics, workshop, science, technology, engineering, mathematics, learning, STEM

Heuristic Potential of Open Institutional Models in Researchers Education

Liudmyla Kryvoruchka, *kryvoruchkald@ukma.edu.ua*
National University of Kyiv-Mohyla Academy, Ukraina

Abstract
The national and cross-border training networks for PhD education established in accordance with EU principles for innovative doctoral training and open science agenda have already contributed to significant transformation of traditional researcher education settings. This poster presents the preliminary results of the Erasmus+ project "Structuring Cooperation in Doctoral Research, Transferrable Skills Training, and Academic Writing instruction in Ukraine's regions" named by DocHub (2016-2019), being implemented in 11 Ukrainian universities and research institutions. The obvious impact of collaborative efforts at organisational and systematic level often hides substantial heuristic potential of diverse instructional and mentoring strategies as well as learners' experiences. The jointly established learning environment of DocHub is structured by the idea of openness and aimed at engaging PhD students and early researchers in active and continuous exchange and thus enhance dialogic and innovative mind-set and support experimentation.

Keywords
Dialogic and innovative mind-set, doctoral studies, inter-institutional cooperation, openness

Construction of a Project Monitoring Application Iteratively and Incrementally

Pekka Mäkiaho, *pekka.makiaho@uta.fi*
Timo Poranen, *timo.t.poranen@uta.fi*
Katriina Vartiainen, *katriina.vartiainen@uta.fi*
University of Tampere, Finland

Abstract
Developing complex software applications is a challenging task. It is hard to recognise all needed features in advance, and therefore software development is often done iteratively and incrementally. Every iteration usually contains activities like planning, refining requirements, implementation and testing. In this paper, we describe how university student teams developed Metrics Monitoring Tool (MMT) application during the years 2014-2018. The construction process of the MMT has contained five larger development iterations so far. All versions have been tested comprehensively with dozens of real users.

Keywords
Iterative and incremental development, metrics monitoring, application

Exploration of Algorithm Abstraction Process with Cubetto and Middle Grade Elementary Kids

Yoshiaki Matsuzawa, *matsuzawa@si.aoyama.ac.jp*
Misako Noguchi, *a8115165@aoyama.jp*
Issei Nakano, *a8115156@aoyama.jp*
Aoyama Gakuin University, Japan

Abstract
In this research, the question of how lower grade elementary school students can solve a programming task which requires algorithm abstraction, was examined using the programming kit “Cubetto” designed for kids. In order to scaffold kids thinking abstraction, the following tools were introduced: (1) the “human-size robot simulator” where kids can simulate their algorithm using their embodied knowledge, and (2) the “Whiteboarding” method where kids can think of an abstraction by finding patterns in a visualized concrete algorithm. In-depth qualitative analysis conducted for the two task-solving episodes illustrates the contrast of the procedures: the one where a young kid could maintain high concentration over thirty minutes with the prepared scaffolds, whereas another where the kids lost interest in their haphazard trial-and-error process.

Keywords
Computational Thinking, Algorithm Abstraction, Cubetto, Lower Grade Elementary Kids

Influence of Students' Self-perceived Use of Metacognitive Strategies and Sensory Preferences on Academic Achievement in Science and Technology

Enric Ortega Torres, *eortega@florida-uni.es*
Florida Universitària, Valencia, Spain

Vincent Sanjosé López,
Joan-Josep Solaz Portolés
Universitat de Valencia, Spain

Abstract
The present work gives empirical evidence of the influence of Secondary students' self-perceived use of learning strategies (including self-efficacy beliefs) and sensory preferences on academic achievement in Science &Technology subjects. MLSQ scales and VARK scores were used as predictors of the average academic marks. A significant percentage of the variance was explained, with a specific contribution from strategies with a metacognitive component.

Our data suggested that about 17% of the variance in academic achievement could be explained by the students' perceptions on their use of learning strategies and individual sensory preferences. When only the strategies with metacognitive basis were considered, the explained variance reached 16%. In particular, main contributions were obtained from Self-Efficacy and Metacognitive Regulation strategies, and also from the Kinaesthetic score from VARK questionnaire.

Keywords
Science and Technology education; MSLQ learning strategies, VARK sensory preferences, Secondary students, Academic achievement

Modeling Across the Subjects

Barbara Sabitzer, *barbara.sabitzer@jku.at*
Johannes Kepler Universität Linz, Austria

Abstract
Jeannette Wing defines computational thinking as fundamental *skill* for everyone. We postulate that modeling is a fundamental *tool* for everyone for constructing knowledge and, certainly, for teaching and learning computational thinking. This is the basis for our new project "Modeling across the Subjects" with two main aims: (1) integrating computational thinking as transversal theme in primary, secondary and teacher education and (2) supporting learning and teaching in different subjects through modeling techniques from the field of computer science (e.g. UML diagrams). This contribution presents the project, its research focus, and some preliminary results from interviews with teachers and students.

Keywords
modeling; computational thinking; learning strategies; brain-based learning

Constructing What? Knowledges of the Powerful, and Powerful Knowledges

Michael Tan, *michael.tan@nie.edu.sg*
National Institute of Education, Nanyang Technological University, Singapore

Abstract
If we consider constructionism as a pedagogical theory, it can be easy to neglect curriculum considerations—what is it we ought to communicate, and why. One approach to this problem is to be deliberate about the nature of knowledge, and the sociological implications of its differential distribution. The epistemologies of the knowledges of Science, Technology, Engineering, and Mathematics (STEM) may be distinguished by its tendencies towards generalisation (S/M), or its contextual application (T/E). Understanding this distinction may be a key to being clear what is being constructed, and how constructionism may be deployed for particular goals. In this theoretical paper, I describe the foundations for such a project.

Keywords
Epistemology; Nature of Science; Nature of Engineering; Curriculum

Towards Girls' Self-perception in Technology and Craft: Challenges and Implications

Sawaros Thanapornsangsuth, Nathan Holbert Monica Chan
{st2839, holbert, mmc2265}@tc.columbia.edu
Teachers College, Columbia University USA

Abstract
This poster explores findings on elementary girls' perception, confidence, and personal experiences in technology and craft. We look into how they define technology and their exposure to technology as consumers and producers. We also asked questions about their experiences with craft materials and activities. Despite coming from families that encourage and financially support extracurricular activities in STEM field, the majority of girls did not feel confident about their knowledge and skills in technology. They also had a tendency to position their own expertise as inferior to their male family members. When asked about crafts the girls showed high interest and engagement and each identified herself as someone who is good with crafts.

Keywords
girls in technology; technology education; craft; technology confidence

Collaborative Creative Music Activity with ICT: A Case Study for Children in Grade Five

Sayaka Tohyama, *tohyama@inf.shizuoka.ac.jp*
Yugo Takeuchi, *takeuchi@inf.shizuoka.ac.jp*
Shizuoka University, Japan

Abstract
How to support learning new knowledge without spoiling children's creativity seems a shared question in both constructionism community and STEAM education. We designed a workshop for 5th graders to scaffold learning musical knowledge using Jigsaw and "Vocaloid for Education" as "object-to-think-with". The result was that fourteen children out of twenty succeeded to make harmonies to given melodies using knowledge which they learned in the workshop even though they had no previous experience of Vocaloid, and five out of the fourteen created unique ones.

Keywords
Music, Vocaloid for Education, Collaborative Learning, STEAM

Applied Constructionism: Critical Reflection and Learning Through Play in Adult Learning

Nalin Tutiyaphuengprasert, *nalin@stanford.alumni.edu*
Darunsikkhalai School for Innovative Learning, King Mongkut's University of Technology, Thailand

Abstract
This poster discusses the combination of constructionism and critical reflection in a workshop design for adults in Thailand. Thailand applied constructionism in different fields including professional development for employees focusing on facilitating self-development and soft skills. In these situations, learners work together to create complex Rube Goldberg machines in less than 24 hours. Time pressure helped learners to reconnect with others and cope with their negative thoughts. While Papert (1999) stated "This acceptance of "negatives" is very characteristic of the Logo spirit", Mezirow's critical reflection (1990) can support safe expression of "negatives" and put them as an "opportunity" to understand and transform.

Keywords
Applied Constructionism, Critical Reflection, Adult Learning, Professional Development, Agency, Team Building, Metacognition, Transformative Learning, Perspectives Transformation

Different Cultures – Different Approaches to Reasoning and Algorithms

Valentina Dagienė, *valentina.dagiene@mii.vu.lt*
Lina Vinikienė, *lina.vinikiene@mii.vu.lt*
Vilnius University, Lithuania

Abstract
The following five strands of mathematical actions are important in school curricula: 1) conceptual understanding; 2) procedural fluency; 3) strategic competence; 4) adaptive reasoning; 5) productive disposition. These strands have implications for mathematics teaching of both the practical and specialised perspectives and are more or less cultural sensitive. One of the challenges facing mathematics educators is to incorporate each of the mathematical strands in a multicultural classroom. We developed the module for teacher educators that provide theoretical background based on the role of reasoning and algorithmic thinking in mathematics. The module involves practical activities based on the understanding of etnomathematics and Seymour Papert ideas. Through this module prospective teacher are able to investigate the understanding different approaches of reasoning and algorithms, explore examples of different approaches, experiment and reflect on the use of tasks, practice different algorithms, develop pedagogical approaches. The module is prepared under Erasmus+ project "Intercultural learning in mathematics and science education, IncluSMe". The project aims to increase the quality of higher education curricula for prospective teachers by linking maths and science education with intercultural learning (http://inclusme-project.eu). The poster describes the module content.

Keywords
Algorithmic thinking, etnomathematics, intercultural earning, reasoning in mathematics

The Web – A Platform for Creation

Márton Visnovitz, *visnovitz.marton@inf.elte.hu*
Győző Horváth, *horvath.gyozo@inf.elte.hu*
Eötvös Loránd University, Faculty of Informatics, Dpt. of Media & Educational Informatics, Hungary

Abstract
The web is a platform that is becoming more and more accessible for many people. With the advancements in web technologies programming on the web is becoming easier to learn. The browser provides a programming API that can be used to express ideas quickly and easily. Using web technologies in learning programming has a lot of advantages, and this platform can be used to apply the constructionist methodology as well. With creative thinking, and basic programming knowledge students can use the web realise their ideas be it a simple computer game, a simulation or any other application. The same programming logic can be applied to many different scenarios; thus, students can explore the way to make their own ideas come to life, learning technology and programming in the process.

Every web application can be represented by three main components. These Three Pillars of a web-based application are: describing the underlying data structure, defining how that data can be displayed in a user interface and how that user interface behaves. Getting started with these steps does not require any technological knowledge about the platform itself, so students can aquire that during the process of implementation.

Keywords
Constructionism, web, JavaScript, game programming

Programming Lessons for Kindergarten Children in Japan

Takeshi Watanabe, *watanabe@viscuit.com*
The University of Electro-Communications, LLC. Digital-pocket, Japan

Yuriko Nakayama, *nakayama@uec.ac.jp*
Kagawa-fujimigaoka Kindergarten, Japan

Yasunori Harada, *hakase@viscuit.com*
LLC. Digital-pocket, Japan

Yasushi Kuno, *y-kuno@uec.ac.jp*
The University of Electro-Communications, Japan

Abstract
Viscuit is a programming language developed by the one of the author. Distinguishing feature of Viscuit is that the programs are made of pictures only, and no characters are required to understand or make programs. This feature makes it possible to experience programming for pre-school children. Actual, LLC. Digital-pocket cooperates with Kagawa-fujimigaoka kindergarten in Japan for regular Viscuit classes since November 2015. In this poster, we analyzed children 's programs and videos taken during the lessons held for them in 2017. In addition, we took questionnaire on the difference between the usual state and the state of the Viscuit lessons of the children to the teacher in charge of each class, and examined the features of the programming lesson at the kindergarten.

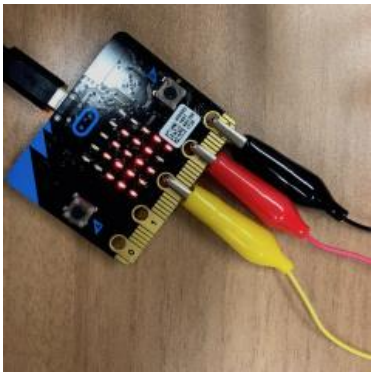
Keywords
Programming for Kindergarten, Visual programming language, Viscuit

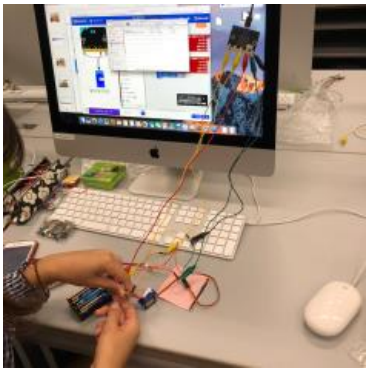
A Practical Report on a Programming Course with "Making" Using Micro:bit

Aoi Yoshida, *aoi@si.aoyama.ac.jp*
Kazunari Ito, *kaz@si.aoyama.ac.jp*

Kazuhiro Abee, *abee@si.aoyama.ac.jp*
Aoyama Gakuin University, Japan

Abstract
We design and practice hands-on course aimed at acquiring problem solving skills and thinking skills in programming through physical computing based on constructionism. We held 15 weeks classes, and there were 28 students who are third-year or fourth-year in this course. This course provided to make products with micro:bit using digital fabrication and physical computing. This poster describes the course content and students' final products.





Micro:bit (Left) Class scenery of this course (Right)

Keywords
Physical computing, Making, Digital Fabrication, Programming education

An Experimental Exploration of the Development of Design Thinking in University Maker Courses

Jinbao Zhang, *Zhangjb@bnu.edu.cn*
Beijing Normal University, Department of Education, China

Abstract
At present, Maker activity in colleges and universities rarely appears in the formal curriculum, and most of them exist in makerspace, training activity of entrepreneurship. In the research, we design a new undergraduate course named as *3D Printing and Maker Education*, including two parts: (1) the application instruction about the technologies of 3D printing, 3D software, Arduino and APP Inventor; (2) design thinking based innovative and creative design practice. In the process of the curriculum, the author does some observation and interview. The primary findings include the design thinking application level in creating scenarios and formation, the bottlenecks and obstacles encountered in various stages of design thinking.

Keywords
Design thinking, Maker, Curriculum design, Instructional design

PANELS / WORKSHOPS / DEMONSTRATIONS / WORKING GROUPS

PANELS

Constructionism across Cultures: Commonalities and Differences of Constructionist Implementations around the World

Paulo Blikstein, *paulob@stanford.edu*
Stanford University, USA

Abstract
The goal of this panel is to bring together scholars from around the world to discuss commonalities and differences amongst implementations of constructionist projects in different countries. Seymour Papert himself refused to precisely define what constructionism was, which opened up the possibility of local contextualization and definition by scholars and practitioners. The papers in this symposium analyze how constructionist theory was used, defined, and implemented within rich cultural contexts, discussing particular characteristics that might have emerged as a result of the combination of local educational practices and philosophies and constructionist theory.

Keywords
Constructionism, cultural context, implementation

Constructionism at Scale

Celia Hoyles, *c.hoyles@ucl.ac.uk*
Richard Noss, *r.noss@ucl.ac.uk*
University College London, UK

Yasmin B. Kafai, *kafai@upenn.edu*
University of Pennsylvania, USA

Kylie Peppler, *kpeppler@indiana.edu*
University of Indiana, USA

Deborah A. Fields, *Deborah.Fields@usu.edu*
Utah State University, USA

Nathan Holbert, *holbert@tc.columbia.edu*
Teachers College, Columbia University , USA

Abstract
Constructionist designers have used new technologies to engage leaners in rich opportunities to build personally meaningful artifacts for decades. In recent years, new technologies have brought these experiences to large numbers of learners in distributed places. In this symposium we bring together expert designers and scholars that have successful developed constructionist innovations using emerging technologies in a range of domains and have brought these innovations to scale. In a panel discussion format, participant highlight key challenges for scaling constructionist design and discuss how these tools and environments evolve as the community of learners increases by orders of magnitude.

Keywords
programming; construction; making; mathematics; scale

Inside the Trojan Horse – A Discussion Among the Next Generation of Constructionists

Sylvia Martinez
Constructing Modern Knowledge, USA

Abstract

"I think the technology serves as a Trojan horse all right, but in the real story of the Trojan horse, it wasn't the horse that was effective, it was the soldiers inside the horse. And the technology is only gong to be effective in changing education if you put an army inside it which is determined to make that change once it gets through the barrier...

Just 100 years ago, John Dewey was saying things about educational change, not very different from what I believe in. He couldn't get very far. And the reason why he couldn't get very far is that he had only philosophical arguments. He didn't have an army. You must have an army, and it's an army primarily of children and the adults also are a political force in this." (Papert 1999)

This discussion, comprised of practicing educators from around the world, will address personal challenges and triumphs bringing constructionism to life on a daily basis. Examples of classroom practice, student projects, professional development, and strategies for sustaining constructionism will be shared.

The panelists:

• Sylvia Martinez – President, Constructing Modern Knowledge (moderator)

• Amy Dugré – Director of Technology, Innovation, and Curriculum: Dusseldorf International School, Germany

• Angela Lombardo – MalpighiLaB & CoderDojo Bologna, Bologna, Italy

• Susana Tesconi - Lecturer and researcher at Department of Information Technology /UOC/ Universitat Oberta de Catalunya DARTS/Interdisciplinary research group in arts, technoscience and society, Barcelona, Spain

• Tracy Rudzitis – Faculty, Constructing Modern Knowledge, USA

• Brian C. Smith – Computing & Learning Technology Teacher: Hong Kong International School, Hong Kong

• Jaymes Dec – Fab Lab Integrator: Marymount School, New York City, USA

Papert, S. (1999). "Ghost in the Machine: Seymour Papert on How Computers Fundamentally Change the Way Kids Learn." Interview of Seymour Papert by Dan Schwartz.

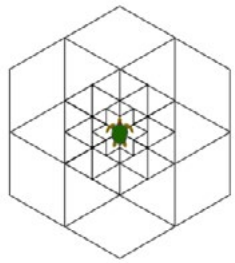
Keywords
constructionism, Logo, Seymour Papert, Scratch, physical computing, coding, teacher education, fabrication

WORKSHOPS

WS1: The Essence of Programming at School – Learning for Life

Jacqueline Staub, *Jacqueline.staub@inf.ethz.ch*
ETH Zurich, Switzerland
Pädagogische Hochschule Graubünden, Chur, Switzerland

Abstract
School is responsible for priming and preparing pupils such that they develop a deep understanding of technology. Computer science education serves a vital role in fostering algorithmic thinking and problem solving skills, as exemplified by programming. We have been introducing primary school pupils to programming in Logo for almost 15 years and thousands of children across Switzerland have learned to program using our curriculum. In this workshop, we give insights into how our curriculum guides pupils to progress individually and how we make pupils building up competence by recovering from programming errors autonomously. This workshop caters towards educators and people interested in programming education.



Algorithmic thinking, constructionism in novice programming, Logo, spiral curricula, modular design, debugging, syntactic errors

WS2: Developing Body Tracking Software with Scratch and Kinect

Stephen Howell, *stephen.howell@ucdconnect.ie*
SMARTlab, University College Dublin, Ireland

Lizbeth Goodman, *lizbeth.goodman@ucd.ie*
SMARTlab, University College Dublin, Ireland

Abstract
In this workshop participants will design, develop and test programs that can be controlled with the user's body. The code is developed with a visual programming language (Scratch) and an infrared body tracking camera (Kinect).

Starting with simple, familiar retro games, and a creative art and music performance application controlled by keyboard and mouse, participants will learn how to add tracking blocks so prescribed movements including individual joint positions, gestures, and stances can be continuously tracked and responded to.

This workshop is appropriate for educators and researchers interested in teaching Computational Thinking by having students develop proprioceptive programs.

Keywords
Visual Programming Languages; Body Tracking; Proprioception; Body Syntonicity

WS3: Group-based Simulation and Modelling: Technology Supports for Social Constructionism

Corey Brady, *corey.brady@vanderbilt.edu*
Vanderbilt University, Nashville TN, USA

Walter Stroup, *wstroup@umassd.edu*
University of Massachusetts Dartmouth, North Dartmouth MA, USA

Tony Petrosino, *ajpetrosino@austin.utexas.edu*
University of Texas at Austin, Austin TX, USA

Uri Wilensky, *uri@northwestern.edu*
Northwestern University, Evanston IL, USA

Abstract
Through a US National Science Foundation-funded project called Group-based Cloud Computing (GbCC), we have created a web-based system for designing and implementing group-centred modelling activities on top of the NetLogo Web platform. For the past two years we have been iteratively building and testing this system in our own classrooms and in the University of Texas's UTeach and UTeach Computer Science programs. In this **workshop**, we introduce GbCC to the Constructionism community, bringing participants to a point where they can create and publish their own activities. Removing barriers to designing one's own group-based activities, we open up the discussion to engage the research and teaching challenges and opportunities for a deeply *social* form of constructionism.

Keywords
Collaborative Learning; Collective Learning; Generative Design; Social Constructionism; Agent-Based Modelling; Participatory Simulations

WS4: AI Programming in Snap!

Ken Kahn, *toontalk@gmail.com*
Department of Education, University of Oxford, UK

Abstract
We have added new blocks to Snap! for speech synthesis, speech recognition, image recognition, and machine learning. These run in a Chrome browser without any extensions or installation. An interactive guide has been developed. Twelve sample programs are available. During the workshop participants will explore these new blocks. They will have the opportunity to modify the sample programs. Familiarity with Snap! or Scratch will be very helpful but not absolutely necessary. While this software can run on tablets and smartphones we recommend you bring a laptop.

More details can be found in the full research paper with the same title in these proceedings.

Keywords
Visual programming; machine learning; block languages; Snap!, AI services; Cloud services; speech synthesis; speech recognition; image recognition

WS5: Developing Affordable STEM Maker Projects with BBC Micro:bits and Microsoft MakeCode

Stephen Howell, *Stephen.Howell@microsoft.com*
Microsoft Ireland

Neeltje Berger, *Neeltje.Berger@microsoft.com*
Microsoft Research

Peter Heldens, *Peter.Heldens@microsoft.com*
Microsoft Netherlands Kevin Marshall,
kevmar@microsoft.com
Microsoft Ireland

Clare Riley, *Clare.Riley@microsoft.com*
Microsoft UK

Abstract
In this hands-on workshop, we will challenge the participants to build 3 projects including wearables and other physical artefacts using affordable microprocessors (BBC Micro:bits) and common arts and crafts materials.

Each project was devised by a multi-disciplinary team of teachers and industry educational technologists in an initiative to develop free, high-quality resources with a depth of content that could be taught in a classroom but breadth of accessibility so that students working at home (without a mentor or educator to guide them) could complete the projects.

Keywords
Constructionism; STEM; Computational Thinking; Inclusive Design; Teacher Training

WS6: NetsBlox: A Constructionist Environment for Creating Distributed Applications

Brian Broll, *brian.broll@vanderbilt.edu*
Vanderbilt University, Nashville TN, USA

Corey Brady, *corey.brady@vanderbilt.edu*
Ákos Lédeczi, *akos.ledeczi@vanderbilt.edu*
Vanderbilt University, Nashville TN, USA

Abstract
NetsBlox is a collaborative learning environment extending Snap! with carefully-selected abstractions that enable students to create distributed applications. It makes distributed programming accessible to young learners using simple yet powerful visual programming primitives, an intuitive user interface and a sophisticated cloud-based architecture. This **workshop** introduces the environment and gives participants hands-on experience with activities that demonstrate NetsBlox's utility for creating multi-player games and client-server applications that access public-domain scientific data sources. Our early work using NetsBlox with students suggests that integrating these connected features into learners' early experiences of programming widens their perspective on the role of computation in both inquiry and broader social life.

Keywords
Block-based programming; visual programming; distributed computing; network programming

WS7: The ER4STEM Repository for Educational Robotics

Annalise Duca, *annalise@acrosslimits.com*
Angele Giuliano, *angele@acrosslimits.com*
AcrossLimits Ltd, Hilltop Gardens, Triq L-Inwkina, Naxxar, Malta

Sofia Nikitopoulou, *sophieniki@ppp.uoa.gr*
Nikoleta Yiannoutsou, *nyiannoutsou@ppp.uoa.gr*
Chronis Kynigos, *kynigos@ppp.uoa.gr*
University of Athens, Athens, Greece

Abstract
Using robotics in education is an engaging method for student motivation towards STEM subjects and more. Teachers, educators and researchers who are newly experimenting with the use of robots in the classroom are all asking a very similar question. "Where can I find inspiration to introduce constructionism in my teaching?", "What can I do to teach my subject using robotics?" The answer to this is "The ER4STEM Repository" which will be full of educational resources, activity plans and suggestions for educators. "The ER4STEM Repository" has been underpinned by the basic pedagogical theory underlying its' design in constructionism. This happens through an Activity Plan Template. This template is aligned with the Repository and provides a generic design instrument that identifies critical elements of teaching and learning with robotics based on theory and practice and is expected to contribute to the description of effective learning and teaching with robotics.

Keywords
repository; educational robotics; STEM; robots; sharing ideas; collaboration; Educational Robotics for STEM; Educational Activities; Constructionism; OER

DEMONSTRATIONS

The Relationship between Computer Programming and English Language Skills

Nevin Akcay, *nevin.akcay@yahoo.com*
Gumuspala Vocational and Technical High School, Istanbul, Turkey

Hulya Avci, *hulyaavci@beykent.edu.tr*
Beykent University, School of Foreign Languages, Istanbul, Turkey

Ali Gungor, *ali.gungor@bau.edu.tr*
Bahcesehir University, Vice President, Istanbul, Turkey

Tufan Adiguzel, *tufan.adiguzel@es.bahcesehir.edu.tr*
Bahcesehir University, Faculty of Educational Sciences, Istanbul, Turkey

Abstract
Computational thinking is a problem-solving method applying computer science techniques and encourages programming or coding skills among children. A programming or coding language is a special language for describing computation and expressing a set of instructions on what tasks a computer needs to execute. There has been an ongoing discussion in the literature whether programming languages with a vocabulary of keywords are based on English. However, there is not much literature on how computer programming affects other aspects of life beyond the skills acquired. In this study, we demonstrated the current literature on the relationship between programming and English language skills and the methods of teaching both languages.

Keywords
computer programming; coding; English language skills; learning

Synthesizing the Mesh: Using Constructible Authentic Representations to Gain Intuitive Understanding of Bayesian Reasoning

Monica Chan, *monica.chan@tc.columbia.edu*
Teachers College, Columbia University, USA

Gary C. F. Lee, *garyleecf@mit.edu*
Massachusetts Institute of Technology, USA

Abstract
In recent years, there has been a massive push towards building basic computer science knowledge and skills at K-12 levels. However, less has been done towards introducing more complex and application-based fields of computer science, such as machine learning, and connecting these fields with mathematics and statistics education. This game *Synthesizing the Mesh* has been developed to provide a constructionist learning platform that helps upper middle to high school students gain intuition and understanding for Bayesian reasoning, an inherent algorithm of machine learning that is traditionally perceived as an advanced topic. The design principle Constructible Authentic Representations (CAR) has been used to guide the game design process.

Keywords
Game Design; Middle School; High School; Constructionism; Problem-solving; Strategy; Bayesian Reasoning; Technology

Teaching Computational Thinking with Minecraft & Microsoft MakeCode

Stephen Howell, *stephen.howell@microsoft.com*
Microsoft Ireland

Abstract
In this demonstration, we present Microsoft MakeCode, an open-source visual programming language that can be used to teach Computational Thinking concepts with Minecraft. With MakeCode, we will show how Minecraft can be used as the microworld for exploring constructionist ideas. Students can design shapes and structures to be built algorithmically. Programmable agents can be instructed to navigate, act and explore, and it all takes place in a virtual world that many students find comfortable and familiar because of the success of Minecraft as a game.

Keywords
Computational Thinking; Microworlds; Visual Programming Languages

Interpolating (and Extrapolating) 3D turtle Programs in Beetle Blocks

Ken Kahn, *toontalk@gmail.com*
Department of Education, University of Oxford, UK

Abstract
Turtle programs can be treated as objects to manipulate. In this demo a program takes two turtle programs as input and creates a new program that is the interpolation between the input programs. An input of .25, for example, will behave like one-fourth of the first program and three-fourths of the second. An input greater than 1 will extrapolate beyond the second program in the direction from the first program. This idea was explored in for two-dimensional turtle programs. Here we generalise it for Beetle Blocks, a 3D version of Snap!

Keywords
Program interpolation; Snap!; Turtle programming; 3D Turtles; Beetle Blocks; Codification

Hedgehog: A Versatile Controller for Educational Robotics

Markus Klein, *klein@pria.at*
Clemens Koza, *koza@pria.at*
Wilfried Lepuschitz, *lepuschitz@pria.at*
Gottfried Koppensteiner, *koppensteiner@pria.at*
Practical Robotics Institute Austria

Abstract
We describe Hedgehog, an educational robotics controller designed to foster interest in STEM subjects. Hedgehog allows building robots out of common actuators and sensors, and can be combined with Lego for a beginner-friendly experience. Through building robots, students can learn and apply engineering, electronics, planning and teamwork skills. The controller facilitates learning programming at different age levels through both textual and visual programming support. For advanced students, Hedgehog's open source ecosystem allows delving into subjects such as microcontroller programming or cooperative robots as well. Hedgehog has been used in numerous workshops and also in robotics competitions with great success.

Keywords
Robotics, Programming, Visual Programming, STEM

WORKING GROUPS

WG1: Constructionist Approaches to Computational Thinking

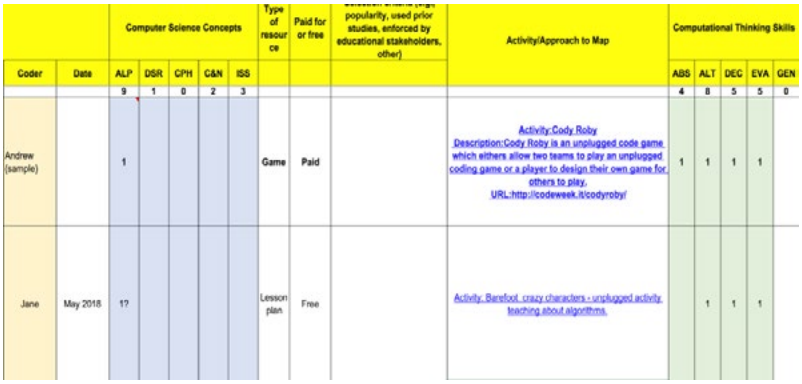
Bernhard Standl, *bernhard.standl@ifs.tuwien.ac.at*
Gerald Futschek, *gerald.futschek@tuwien.ac.at*
Vienna University of Technology, Austria

Jane Waite, *jane.waite@computingatschool.org.uk*
Queen Mary, University of London, UK

Andrew Paul Csizmadia, *A.P.Csizmadia@staff.newman.ac.uk*
Newman University Birmingham, UK

Abstract

This working group elaborated an approach to evaluate computational thinking activities with learning in a constructionist way. As widely known, Wing stated in her refined definition of Computational Thinking (CT) , that CT is an approach for solving problems that draws on concepts fundamental to computing. Later, Aho described the term CT as including algorithm-design and problem-solving techniques that can be used to solve common problems arising in computing. As Yadav et. al. reminded Wing's initial paper points out that CT involves three key elements Algorithms, Abstraction, and Automation. The term CT has been grown since then to a variety of interpretations. Ackerman compared Piaget's constructivism and Papert's development of this in a constructionist way and drew the two views together as learning in a constructionism way 'as Piaget and Papert do, that knowledge is actively constructed by a child in interaction with its world, then we are tempted to offer opportunities for kids to engage in hands-on explorations that fuel the constructive process'. Papert's core message that the learner is 'projecting out our inner feelings and ideas is a key to learning. Expressing ideas makes them tangible and shareable which, in turn, informs, i.e., shapes and sharpens these ideas, and helps us communicate with others through our expressions.' This means, that new insights are the sum of single experiences made by applying existing knowledge for enhancing it. Considering both parts discussed above, constructionism and computational thinking, this working group's intentions are based on the combination of both for selecting and evaluating classroom activities. Therefore, we designed a matrix, where aspects from both, computational thinking and a constructionist learning approach, can be analysed. The matrix is designed to identify, categorize and evaluate such classroom activities and encompasses three parts: Computer Science Concepts, Problem-Solving Concepts, Levels of Abstraction.



		Computer Science Concepts					Type of resource	Paid for or free	popularity, used prior studies, enforced by educational stakeholders, other)	Activity/Approach to Map	Computational Thinking Skills				
Coder	Date	ALP	DSR	CPH	G&N	ISS					ABS	ALT	DEC	EVA	GEN
		9	1	0	2	3					4	8	5	5	0
Andrew (sample)		1					Game	Paid		Activity:Cody Roby Description:Cody Roby is an unplugged code game which either allow two teams to play an unplugged coding game or a player to design their own game for others to play. URL:http://codeweek.it/codyroby/	1	1	1	1	
Jane	May 2018	1?					Lesson plan	Free		Activity: Barefoot crazy characters - unplugged activity teaching about algorithms.		1	1	1	

This working group's paper presents our approach of a systematic evaluation of classroom activities in terms of constructionist learning and discusses first results of our evaluation process, where we coded such activities corresponding to the matrix.

Keywords

Computational Thinking, Constructionist Learning, Classroom Activities

WG2: Developing Constructionism, or a New Learning Concept, across the Ages

Don Passey, *d.passey@lancaster.ac.uk*
Lancaster University, Lancaster, UK

Loice Victorine Atieno, *atienomunira04@gmail.com*
Eötvös Loránd University, Budapest, Hungary

Wilfried Baumann, *baumann@ocg.at*
Austrian Computer Society, Vienna, Austria

Valentina Dagienė, *valentina.dagiene@mii.vu.lt*
Vilnius University, Vilnius, Lithuania

Abstract
Curricula in many countries are adopting computing (or informatics) as a subject or discipline. Computing learning practices in these 'new' curricula can involve pupils from 5 years of age, either within a discrete subject, or integrated into other subject topics across the curriculum. However, the learning concept, framework or theory that such curricula are based on, is not clear in curriculum documentation.

Our curriculum concepts of learning progression are largely based on Piaget's research, who described learning as a form of cognitive constructivism, developing over the age span of young people, and progressing through a series of stages: sensorimotor; preoperational; concrete operational; and formal operational. The new computing curriculum and teaching and learning practices could well be placed within this conceptual framework. However, Vygotsky's research added a more social dimension of learning, a concept of social constructivism.

Our contribution will review the constructionism approach by Papert that is based on uses of digital and computing-based resources within a constructivist approach to learning, and will investigate what learning concept, framework or theory should underpin computing curricula to ensure that practices and outcomes are as effective as possible.

Keywords
Constructivism, Activity theory (Lev Vygotsky), Learning Theories

WG3: Creating and Looking at Art with Logo Eyes

Evgenia Sendova, *jenny.sendova@gmail.com*
Institute of Mathematics and Informatics at the Bulgarian Academy of Sciences, Bulgaria

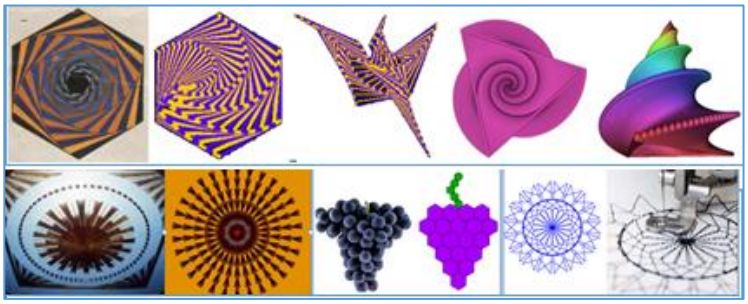
Christos Chytas, *christos.chytas@uni-oldenburg.de*
Computing Education Research Group, University of Oldenburg, Germany

Katarzyna Olędzka, *katarzyna.oledzka@oeiizk.waw.pl*
Computer Assisted Education and Information Technology Centre in Warsaw, Poland

Ralf Romeike, *ralf.romeike@fau.de*
University of Erlangen-Nuremberg, Germany

Wolfgang Slany, *wolfgang.slany@tugraz.at*
Catrobat Team, Graz University of Technology, Austria

Abstract
Computation is an integral component in nearly every dimension of our lives and *provides new tools for self-expression even in areas that one would not dare to enter without the access to computers.* In 1967, Seymour Papert, Wally Feurzeig and Cynthia Solomon designed Logo to allow children and youth to explore programming and mathematics concepts, while providing exciting opportunities for creativity and experimentation. Computation provides an appropriate framework for dealing with notions of *how to* and *what if* rather than with notions of *what is*, which already has an impact on the way we teach fine arts, music, or literature. For Papert and his fellow constructionists the programming was not only a technical skill for industry but also a mode of creative and personal expression. Today, the absence of computation in mainstream art and science education is still apparent but the appearance of Logo and new modern tools that inherit the ideas of the Logo educational philosophy (known as *constructionism* today) promise to change that. Drawing on examples from experiences with IT teachers and children in participatory design courses, we discuss *how to harness the Logo culture of "making things happen, making things work" so as to open the eyes of young people to the beauty around them.*



Artistic stylization in Logo style

Keywords
Art; Creativity; Coding; Logo; Visual Programming; Maths; Stitching; Embroidery

WG4: The Constructive Strategies in Teaching Humanities with Films[6]

Lilija Duoblienė, *lilija.duobliene@fsf.vu.lt*,
Jūratė Baranova, *jurabara@gmail.com*
Vilnius university, Lithuania

Luc Anckaert, *luc.anckaert@kuleuven.be*
KU Leuven, Belgium

Wilfried Baumann, *baumann@ocg.at*
Austrian Computer Society, Wien, Austria

Abstract
The main research question: is it possible to apply the constructive methods of education in teaching humanities with films. The skeptics suppose: it is not possible for several reasons. The article starts the discussion, suggests some arguments for justification of positive answer and reveals several different possible experiences of including constructionism in teaching humanities with films. The theoretical premises for these approaches are the pragmatic pedagogy of John Dewey, Deleuzean theory of cinema and Paul Ricouer's theory of personal identity

The authors discovered and suggested three possible ways of using constructive strategy in the process of teaching with films: 1. Creation of the students their own multimodal projects, trying to find proper images for their ideas; 2. Watching films alongside with reflection of some philosophical concepts, afterwards writing essays and presenting them to the group in the discussion as the possibility to encounter the glimpses of reality from the different specter of perspectives; 3. Using film as a challenge to purify student's own mimetic processes by placing them in a broader metaphorical context. The film in this case is the imaginative-metaphorical construction of the personal identification process. The authors discuss the three strategies as parallel, not opposing or excluding each other. All three strategies lead to the constructing students' world view not with very concrete separate elements of clear shape and content, but with elements of absolutely different level, plane, from the different assemblages and enlarge the capacity of their critical and creative thinking. Also these approaches develop their social capacities – the ability of the understanding and communication with the different other.

Keywords
constructionism, humanities, teaching with films, Deleuze, film theory, Personal identity

[6] The half of the article is based on the investigation included into the project "Philosophical Sources and Problems of Multimodal Education" financed by the Research Council of Lithuania (No.S- MIP-17-37)

WG5: Constructionism in the Classroom: Creative Learning Activities on Computational Thinking

Michael Weigend, *mw@creative.informatics.de*
Holzkamp Gesamtschule Witten, University of Münster, Germany

Zsuzsa Pluhár, *pluharzs@caesar.elte.hu*
Eötvös Loránd University, Faculty of Informatics, Budapest, Hungary

Igor Pesek, *igor.pesek@um.si*
University of Maribor, Faculty of Natural Sciences and Mathematics, Slovenia

Anita Juškevičienė, *anita.juskeviciene@mii.vu.lt*
Vilnius University Institute of Data Science and Digital Technologies, Lithuania

Jiří Vaníček, *vanicek@pf.jcu.cz*
University of South Bohemia, Faculty of Education, Czech Republic

Kazunari Ito, *kaz@si.aoyama.ac.jp*
Aoyama Gakuin University, Japan

Abstract
This contribution discusses learning activities without computer ("unplugged activities") related to computational thinking, which challenge creativity. These arrangements have these common properties: (1) The students create a product that can be shown around later. This may be a physical artefact or a performance, which could be documented (photo, video). The concrete outcome is very individual and may be surprising in contrast to analytical tasks with just one correct solution. (2) The activity is experience-based. It (ideally) demands all the senses and challenges the whole person. The students do not use a computer ("unplugged") and do not develop a program but may use Lego blocks, pencil and paper or other material found in their environment. (3) The activity can be performed without preparation ad hoc in one lesson in 5 to 40 minutes, in contrast to projects that are carefully planned and require weeks of work.

Based mostly on Csikszentmihalyi's system model of creativity, this contribution discusses the possibilities of being creative in the classroom. Extending Csikszentmihalyi's approach, the classroom situation is seen as a "local" creative system. Specifics of the domain "computational thinking" are discussed.

This study distinguishes between four types of creative unplugged activities:

- Create an algorithm solving a given problem and present it.
- Create to a given algorithm or informatics concept a new situation, in which this algorithm or concept can be used as well and present it.
- Create an algorithm with certain structural properties (like loops, recursion, functions calls etc.) and present it.
- Create a visualisation for a given algorithm or concept of computer science and present it.

More than 300 computer science educators (school and university) from different countries were asked about this classification scheme, about their experience with creative unplugged activities and their opinion about relevance and educational potential of the different types of activities. The results of qualitative and quantitative analysis of the answers suggest that the classification scheme is quite acceptable for the community of CS educators. Creative unplugged activities are not often used in CS education but they are considered to be relevant.

Keywords
Computer science unplugged, creativity, computational thinking

WG6: Learning to Program in a Constructionist Way

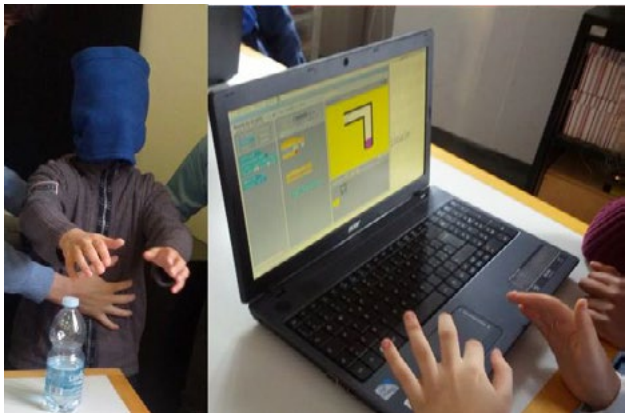
Mattia Monga, *mattia.monga@unimi.it*
Università degli Studi di Milano, Milan, Italy

Michael Lodi, *michael.lodi@unibo.it*
Alma Mater Studiorum - Università di Bologna & INRIA Focus, Italy

Dario Malchiodi, *dario.malchiodi@unimi.it*
Anna Morpurgo, *anna.morpurgo@unimi.it*
Università degli Studi di Milano, Milan, Italy

Bernadette Spieler, *bernadette.spieler@ist.tugraz.at*
Technische Universität Graz, Austria

Abstract
Constructionism is a strategy of education which has its roots in Piaget's constructivist theory of learning as an active process, in which people actively construct knowledge from their personal experience of the world. In general, students do not just receive pre-built ideas from teachers: they have to make them up by engaging themselves with problems, projects, and other people (instructors, but also peers). Papert's constructionism indeed emphasizes the importance of having personally-meaningful goals and "public artifacts" (not necessarily concrete ones: either "a sand castle on the beach or a theory of the universe") that can be shared and discussed with others interested in the same (learning) enterprise. This is sometimes summarized with the four P-words: Projects, Peers, Passion, Play and this motto indeed inspired successful educational initiatives such as the Scratch programming language. However, while programming is often seen as a key element of constructionist approaches (starting from Papert's Logo, a programming language designed to enable the learning of geometry), the research on learning to program through a constructionist strategy is somewhat limited, mostly focusing on how to bring the abstract and formal nature of programming languages into "concrete" or even tangible objects, graspable even by children with limited abstraction power. Notwithstanding this, constructionist ideas are floating around mainstream programming practice and they are even codified in some software engineering approaches: agile methods like eXtreme Programming, for example, suggest several techniques that can be easily connected to the constructionist word of advice about discussing, sharing, and productively collaborating to successfully build knowledge together; moreover the incremental and iterative process of testing ideas fits well with the agile preference to "responding to change over following a plan".



Keywords
programming, constructionism

WG7: Constructionism in Upper Secondary and Tertiary Levels

Ana Isabel Sacristán, *asacrist@cinvestav.mx*
Center for Research and Advanced Studies (Cinvestav), Mexico

Richard Akrofi Kwabena Baafi, *richbaafius@yahoo.com*
Eotvos Lorand University, Hungary

Lina Kaminskienė, *l.kaminskiene@pvt.vdu.lt*
Vytautas Magnus University, Lithuania

Mihael Sabin, *Mihaela.sabin@unh.edu*
University of New Hampshire, USA

Abstract
In the constructionist paradigm, the fundamental premise is to create student-centered learning situations for students to consciously engage in constructing shareable, tangible objects, through meaningful projects. In Papert's vision, one particularly valuable means of doing that is in programming the computer because, in doing that, the student "establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building" (Papert, 1980, p. 5). Since the 1980s, there have been countless experiences and studies exploring and documenting the use of the constructionist paradigm, many of the first ones using Logo computer programming, but mostly with young students at primary or middle-school levels. However, experiences in upper secondary and university levels are scarcer. Laurillard (2002), nonetheless, advocates for constructionist and collaborative technology-based learning environments in higher education; she says (p. 42): "the aim of university teaching is to make student learning possible [...] not simply impart decontextualised knowledge, but must emulate the success of everyday learning by situating knowledge in real-world activity" helping students reflect on their experience of the world and ways of representing it.

The purpose of this working group was to share constructionist experiences in upper secondary and tertiary educational levels, particularly those involving computer programming and/or computational thinking and environments; and to reflect on the challenges, needs and differences of constructionist technology-based implementations in the various educational levels, and on how to promote such implementations in upper levels.

The guiding and research questions were:

- What are the characteristics of constructionist implementations in upper educational levels? In upper secondary school? At university level? How are they different from lower levels? What are the particular challenges?
- How is, or what could be, the role of digital technologies and computer programming in such implementations?
- How can constructionist implementations be integrated and promoted in higher education? What is required for that?
- Can real-life data, phenomena and problems be harnessed for developing such implementations?

References
Laurillard, D. (2002). *Rethinking university teaching: A conversational framework for the effective use of learning technologies*. 2nd Edition. Routledge.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York, NY: Basic Books.

Keywords
Constructionism; upper secondary level; tertiary level

TECHERS' DAY

WS1: Dynamic Teaching Ideas for teaching Music Theory

Judith Bell, *jbell@chisnallwood.school.nz*
Chisnallwood Intermediate School, Christchurch, New Zealand

Abstract
In this workshop you will try out a range of effective, outside-the-square, tried-and-true theory activities based on Judith's creative work with the very popular Chisnallwood School Theory Club. You will learn about music theory games are suitable for a range of ages and easily adaptable for different levels and music abilities, but ideally from ages 7 – 12. Some activities use digital technology, while others are completely unplugged. They are suitable for use in groups – even of mixed levels – and you'll be amazed at the learning that comes out as well as the amount of fun.

WS2: Computer Science Unplugged for Teachers

Tim Bell, *tim.bell@canterbury.ac.nz*
University of Canterbury, New Zealand

Abstract
This workshop will demonstrate ideas from the "Unplugged" approach to teaching Computational Thinking topics. This approach provides opportunities for students to encounter the great ideas in computer science away from computers, and is a useful complement to classes that focus on "coding". We will look at how you can engage your own students with the material, and also "plugged in" activities, which connect the activities to programming. You will learn about free resources that you can use in your classroom immediately, and also how you can integrate these activities with other school subjects. The workshop will be suitable for both primary and secondary educators.

WS3: Developing Algebraic Habits of Mind in Students

Paul Goldenberg, *pgoldenberg@edc.org*
Education Development Center, Waltham, MA, USA

Cynthia J. Carter, *CCarter@rashi.org*
The Rashi School, USA

Abstract
Participants will see one puzzle-centric approach (elaborating on the work of mathematician W.W. Sawyer) aimed at developing the language and logic of algebra in grades 5 and up. Most of the focus of the workshop will be on the beginnings of that algebraic development, but we will also spend some time looking at activities - from paper-ripping to factorials of negative numbers - that help build students' comfort with the mathematical idea of extension, extending ideas that have "natural" meanings in the natural numbers and extending them to "unnatural" places, like negative numbers and fractions. Participants will be encouraged to pose problems of their own, extending problems presented in the workshop, and will receive a packet of materials they can adapt and use with their classes.

WS4: Puzzles & Programming to Develop Mathematical Habits of Mind in 6–10 year Olds

Paul Goldenberg, *pgoldenberg@edc.org*
Education Development Center, Waltham, MA, USA

Cynthia J. Carter, *CCarter@rashi.org*
The Rashi School, USA

Abstract
This workshop will focus on tested work using a puzzle-centric primary mathematics curriculum with an algebraic focus(inspired by mathematician W.W. Sawyer), and new work we are doing now to infuse that curriculum with programming (inspired by work by Jenny Sendova and others in Bulgaria and by the ScratchMaths work done by Richard Noss and Celia Hoyles of the UK and Ivan Kalaš of Slovakia). Puzzles help develop children's logic; programming provides an extra language to help them express that logic. Expressing and experimenting with one's mathematical ideas helps develop them further. Participants will solve and create puzzles that are suitable for young students and will see how computer programming by young students can support their mathematical learning. All materials that will be used or described are available free.

WS5: Powerful Ideas in Lower Primary Programming: High Time to Recognize Them

Ivan Kalaš, *kalas@fmph.uniba.sk*
Comenius University, Bratislava, Slovakia

Abstract
We think that using any of the dozens of existing programming environments in regular primary school setting, with general primary teacher does not work well, mostly because of the following reasons: (a) these environments are not designed for collaborative constructivist teaching where the teacher has to support everybody in the class, and (b) these environments often neglect basic, simple, but key important powerful ideas that pupils should discover and adopt before computational constructs that are considered in general as introductory (sequence, selection, repetition, variables...)

In the workshop we will use our latest development of Emil, a new programming environment and its systematic pedagogy constructed and trialled in a group of design primary schools, with pupils aged 8. Together with the participants we will experience, identify and discuss several powerful ideas that pupils should experience before all other powerful ideas that we have exploited in our ScratchMaths curriculum.

WS6: Snap! - Beauty & Joy of Computing (visually)

Witek Kranas, *witek@oeiizk.waw.pl*
OEIiZK, Poland

Abstract
Goals: familiarise new visual programming environment, reflect some examples of functional programming in Snap!
Expected outputs: some experience with new environment – Snap!, some useful for computer science course examples of programming in Snap!
What we will be doing:
• Acquaint Snap! similar to Scratch environment for visual programming.
• Share and discuss some examples in Snap! from "Beauty and Joy of
• Computing" Berkeley, ECD CS course for 9-12 grade students in US.
• Discuss possible benefits and limitations of visual programming.

WS7: ViLLE – E. Learning Path for Mathematics and Programming

Mikko-Jussi Laakso, *milaak@utu.fi*
University of Turku, Finland

Abstract
ViLLE is an exercise-based education environment that enables easy learning and teaching of mathematics, programming and other topics. The development is research-based, and the features and the methodology utilized have been thoroughly studied with various setups in the Centre for Learning Analytics at University of Turku. For students, ViLLE offers more than 15 000 carefully designed, motivating and activating exercises for learning mathematics and programming. All exercises are automatically assessed and provide immediate feedback. For teachers, ViLLE provides comprehensive learning analytics that visualize everything you need to know about your students' learning process – including automatic detection of misconceptions and real-time analysis of students' progress. ViLLE is used to transfer one lesson a week into an electronic learning experience. There are existing exercises and materials for all nine grades of primary and middle school. Moreover, there are programming exercises integrated into all levels to give students a head start in learning computational thinking and basics of computer science.

Using ViLLE provides evidence-based, scientifically proven results for all grades. In the studies conducted in Finland – the country that excels the Pisa assessments each year – it was confirmed that the students using ViLLE improve their learning significantly more than the control group learning mathematics with traditional pen and paper method. With matching skill levels before the experience, groups using ViLLE achieved at least 20 percent higher scores in the exams conducted at end of the school year. Students using ViLLE also make 70 % less errors than students in a control group. Moreover, the students find ViLLE as highly motivating and fun tool to use.

WS8: Constructionism in Action: Do we Need to Start from Scratch?

Evgenia Sendova, *jenny.sendova@gmail.com*
Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Bulgaria

Nikolina Nikolova, *nnikolova@fmi.uni-sofia.bg*
Faculty of Mathematics and Informatics, Sofia University "St. Kliment Ohridski", Bulgaria

Abstract
Children in the digital era are surrounded by information and communication technologies. The development of digital competences and more specifically –the ability to express their creativity through computational thinking, is evaluated by the society as vital for the contemporary society. This makes it natural to introduce programming courses for students of younger age, new curriculum and even a new school subject – computer modeling.

Should the programming be taught per se though? Does the introduction of new syllabus put a threshold and ceiling on the performance of the teachers? What about programming languages with no threshold and no ceiling (the ceiling being only the user's imagination)?
Bulgaria has a long-term experience in teaching programming, and even better – in learning through programming. The Logo philosophy promotes the programming as a means for learning and creative self-expression. It is in harmony with the family of contemporary programming languages, successors to Logo and developed specially for children.

There will be no threshold for the participants. However, this does not mean that we would start from scratch. Rather, we will start with the traditions of the Logo philosophy and Logo culture in international setting, we'll present the potential of their development through Scratch and of course, we'll work, create and have fun together! Most importantly, we'll rely on high enough ceiling!

WS9: Teaching Coding and Physical Computing

Gary S. Stager, *gary@stager.org*
Constructing Modern Knowledge, USA

Abstract
Learn how a project-approach to computer programming, robotics, and physical computing can serve a diverse student population while developing your own skills. This workshop will explore powerful ideas from computer science and engineering that may be employed in the solving of problems across the curriculum. A review of software and hardware options will be explored in addition to two focused programming and robotics activities. Participants will also have experience with Hummingbird Robotics, the BBC Micro:bit and other low-cost "microcontroller development boards" offering great potential for learning through making, tinkering, and engineering in the classroom.

WS10: The Essence of Programming at School – Logo in a Spiral Curriculum

Jacqueline Staub, *Jacqueline.staub@inf.ethz.ch*
ETH Zürich, Pädagogische Hochschule Graubünden, Chur, Switzerland

Abstract
School is responsible for priming and preparing pupils such that they develop a deep understanding of technology. Computer science education serves a vital role in fostering algorithmic thinking and problem solving skills, as exemplified by programming. This form of learning is constructive, enriches creativity and teaches precision. We have been introducing primary school pupils and their teachers to programming in Logo for more than a decade and thousands of children across Switzerland have learned to program using our curriculum and purpose-built programming environment. In this workshop, we give insights into how our curriculum guides pupils to progress individually and how we make pupils building up competence by recovering from their programming errors autonomously. This workshop caters towards educators and people interested in how to introduce computer science to novices. Participants gain practical insights into our curriculum and discuss its didactic structure.

WS11: How to Create and Sustain a Progressive Pedagogy in a Traditional Setting (Roundtable Discussion)

Carol Sperry Suziedelis, *Carol.Sperry@millersville.edu*
Millersville University, USA

Abstract
Format of the workshop: We will encourage a vibrant discussion of ideas, efforts, questions, and fears about creating dynamic and engaging projects as well as the nurturing of an atmosphere of free expression and exploration in the classroom. Without the support of each other, it may be difficult to relax and let go of the "controls" many think are necessary for a workable classroom. However, students and teachers must have the latitude to pursue the construction of their own knowledge, the freedom to make mistakes and "recalibrate," and the time to discuss the powerful effects of the process. We will consider ways to do it all within the constraints of the average classroom setting.

Along the way, we will revisit some of Seymour Papert's profound insights into the remarkable endeavour of teaching and learning.

I believe the prediction of "expected outputs" is usually precarious for a constructionist teacher. To "expect" certain outputs is, to my mind, a way of limiting them. So, I would say, I hope that we have a lively discussion that inspires ideas, creates new relationships, and bolsters the courage needed to follow a joyful and innovative path.

WS12: Joyful Learning of Geometry in Cultural Context. Analysis and Construction of Geometric Ornaments

Igor Verner, *ttrigor@technion.ac.il*
Technion – Israel Institute of Technology, Israel

Khayriah Massarwe, *massarwe@technion.ac.il*
Technion – Israel Institute of Technology, The Arab Academic College for Education, Israel

Daoud Bshouty, *daoud@technion.ac.il*
Technion – Israel Institute of Technology, Israel

Abstract
Mathematics education seeks to accommodate pedagogical approaches that enhance learning mathematics and make it relevant to today's students who prefer hands-on visual and joyful activities, and are socially inclined. The challenge of teaching is to expose students to the interconnection between real world practices and culturally rooted mathematical ideas. The subject Geometry is unique in its combination of intuitively rooted figural concepts and abstract logical statements. The geometry teacher should facilitate the learner to acquire the language of geometric terms and the ability to use this language correctly when communicating geometric ideas. We have been developed two courses that introduce prospective and in-service teachers to teaching and learning geometry in cultural context using the ethnomathematical approach. The teachers analyzed and constructed, by compass and straightedge, geometric ornaments from different cultures, posed and solved geometric problems related to the ornaments. In this workshop, we aim to engage the participants in constructing and analyzing of culturally meaningful geometric ornaments. The participants will solve geometric problems related to the ornaments and develop new problem by themselves. Then we will discuss the arguments for introducing activities with geometric ornaments in school geometry learning.

VILNIUS UNIVERSITY
THE OLD CAMPUS

ISBN 9 786099 576022

Constructionism
2018, Vilnius

Constructionism, Computational Thinking
and Educational Innovation

August 20-25, Vilnius, Lithunia

Book of Abstracts

Edited by
Valentina Dagienė
Eglė Jasutė