Constructionism 2018
Constructionism, computational thinking and educational innovation
Vilnius, Lithuania, August 21 to 25

Working Group

## Learning to program in a constructionist way

Proposed by **Mattia Monga**, Universitàdegli Studi di Milano, Italy, mattia.monga@unimi.it

Constructionism [PH91] is a strategy of education which has its roots in Piaget's constructivist theory of learning as an active process, in which people actively *con- struct* 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 and projects. 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" [PH91]) that can be shared and discussed with others interested in in the same (learn- ing) enterprise [Res96]. This is sometimes summarized with four P-words: Projects, Peers, Passion, Play and this motto indeed inspired successful educational initiatives such as the Scratch programming language [Res14]. 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 fo- cusing on how to bring the abstract and formal nature of programming languages into "concrete" or even tangible objects, graspable even by children with limited abstrac- tion power [RMMH$^+$09, HJ07, HAA17]. However, constructionist ideas are floating around mainstream programming practice and they are even codified in some soft- ware engineering approaches: agile methods like eXtreme Programming [BA04], for example, suggest several techniques that can be easily connected to the constructionist word of advice about discussing, sharing, and productively collaborating to success- fully build knowledge together [Res96]; moreover the incremental and iterative process of testing ideas [Res07] fits well with the agile preference to "responding to change over following a plan" [BBvB$^+$01].

This working group will study the use of a constructionist strategy to learn to program by considering the multifaceted skills needed by software projects:

- understanding automatic interpreters able to manipulate digital information;
- predicting concrete semantics of abstract descriptions;
- thinking about problems in a way suitable to automatic elaboration;
- devising, analysing, comparing solutions;
- adapting solutions to emerging hurdles and needs;
- organizing team work and productively sharing abstract knowledge.

**References:**

[BA04] Kent Beck and Cynthia Andres. *Extreme Programming Explained: Em- brace Change (2Nd Edition)*. Addison-Wesley Professional, 2004.

[BBvB$^+$01] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for agile software development. http://agilemanifesto.org/iso/en/ manifesto.html, 2001.

[HAA17] Matthias Hauswirth, Andrea Adamoli, and Mohammad Reza Azad- manesh. The program is

the system: Introduction to programming without abstraction. In *Proceedings of the 17th Koli Calling Interna- tional Conference on Computing Education Research*, Koli Calling '17, pages 138–142, New York, NY, USA, 2017. ACM.

[HJ07] Michael S. Horn and Robert J. K. Jacob. Designing tangible program- ming languages for classroom use. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, TEI '07, pages 159– 162, New York, NY, USA, 2007. ACM.

[PH91] Seymour Papert and Idit Harel. *Constructionism*, chapter Situating constructionism. Ablex Publishing Corporation, 1991.

[Res96] Mitchel Resnick. Distributed constructionism. In *Proceedings of the 1996 International Conference on Learning Sciences*, ICLS '96, pages 280–284. International Society of the Learning Sciences, 1996.

[Res07] Mitchel Resnick. All i really need to know (about creative thinking) i learned (by studying how children learn) in kindergarten. In *Proceedings of the 6th ACM SIGCHI Conference on Creativity &Amp; Cognition*, C&C '07, pages 1–6, New York, NY, USA, 2007. ACM.

[Res14] Mitchel Resnick. Give p's a chance: Projects, peers, passion, play. In *Constructionism and creativity: Proceedings of the Third International Constructionism Conference. Austrian Computer Society, Vienna*, pages 13–20, 2014.

[RMMH+09] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosen- baum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: Pro- gramming for all. *Commun. ACM*, 52(11):60–67, November 2009.