Outreach in Computing Education: a Design Toolkit

Agnese Addone
Dipartimento di Informatica, Università degli Studi di Salerno
addone@unisa.it

Abstract

In our society, learners face the constantly growing world evolution, its technological progress and the big questions that innovations spark. The increasing demand of computing competences reflects the spread of targeted training to broaden the cultural and technical instruction. Computing Education supplies the theoretical foundation to these formative objectives, in order to ensure the achievement of a specific knowledge, skills and digital citizenship. Besides formal institutions and their curricula, there is also a complementary world of Computing Education Outreach Programs, led outside school in non-formal and informal learning environments. The contribution of my thesis fits into the perimeter of these initiatives, especially those addressing the students, ranging from primary until the end of high school and before accessing the post-secondary pipeline, also called K-12. I will provide a proposal of a design toolkit to support Outreach Programs designers in K-12 Computing Education, both in the instructional design stage and in the analysis and resolution of the most common critical issues.

Technology is shaping our lives at an increasing rate and is modeling the way we live in our world. Communications, social media, automation, transactions, video games, are just some of the main purposes in which technologies take form. In this context, the demand of specific training in the field of Computer Science is growing, and it allows an aware and competent attitude in addressing the contents and variety of these technologies.

Computing Education is the discipline and research field that deals with these formative needs, from Computational Thinking to advanced Information Technology literacy, from professional skills to the balancing of the digital divide issues in order to ensure a civic participation in the society.

The school system has begun to respond to these cultural and educational needs from a few decades, but there is still no widespread teaching on a global level and the gap is reflected in cultural, economic and social development. Although some countries have started to elaborate curricula in Computer Science, many others have not yet established formal programs of Computing Education.

To compensate the lack of specific pathways, however, many extracurricular initiatives were born and are still widespread, which often act even as a driving force for change in the school: Outreach Programs are non-formal and informal sessions where kids can access computing literacy in an environment designed to meet their motivations towards computing and the major education instances from our society.

Even though researchers and educators, as outreach designers, refer to the contents of the school curriculum in the choice of topics and contents, however these sessions remain complementary to the formal education system in terms of their peculiar features (motivation, organization, teaching principles, objectives and strategies) which differentiate them from the school instruction and organization.

Dissertation contributions and method. My dissertation topic fits in the context of the Computing Education Outreach Programs (CEd OP) for primary and secondary school learners (also called K-12 with a US notation), with the aim to describe the design process and to implement a toolkit to support researchers, educators and practitioners to overcome the major

issues. Many concerns can suggest designers a set of diverse options when elaborating a program proposal or while re-designing the outreach during the follow-ups. Therefore my research dissertation is based on two research questions (RQs):

- **RQ1** - What are the major design concerns of an Outreach Program in K-12 Computing Education?
- **RQ2** - What are the possible solutions to pursue for ensuring efficacy to the Outreach Programs?

In reply to RQ1, I intend to classify the designers' process and concerns by creating a few categories of related issues and areas of influence; by answering RQ2 instead I mean to give a tangible support to researchers in the design or re-design of the programs by providing them a practical toolkit.

In addition to the specific contribution in the elaboration of the design process and toolkit, the thesis deals at a theoretical level with the epistemological problems of the subject [20]. In fact, Computing Education is a relatively recent subject that lies between Computer Science (CS) and Pedagogy, whose respective experts come from both disciplinary fields by hybridizing and merging their knowledge and skills [20].

In the dissertation I have taken into account a research method that I can summarize as follows:

- an *excursus* on the history of Computing Education Outreach Programs and on the major scientific contributions to the inter- and cross-disciplinary link between Computer Science, Pedagogy and Instructional Design;
- an analysis and taxonomy I conducted on the research topic, both on the Outreach Programs [2], and the languages, tools and environments (TLE) [3];
- a description of the Outreach design process and the method to detect the major concerns;
- a design toolkit, with guidelines for designers on possible solutions to the above concerns.

Background and related topics. The first computer language expressly designed for educational purpose, LOGO, date back to the 1967 [19] and, during the same period we find the first experiences in teaching and learning computing with children [16].

The further definition of K-12 Computing Education has had a development over the last twenty years, with studies ranging from school to university, to teachers' professional development [7]. Currently, interesting overviews which delineate the diversity of the K-12 out-of-school programs are currently offered by a few works [4, 13], also in the form of Systematic Literature Reviews on the initiatives [9, 2].

Related works on K-12 Computing Education programs highlight the role of design in planning initiatives that promote crucial skills, such as problem solving, or that underline the evaluation of the artifacts [15, 17]. In many studies, the main pedagogy is student-centered and emphasizes the active role of the learner in the process. The efficacy of Outreach Programs is also related to the role of positive incentives or gratifications, and to the social context. In one of his well-known reflections, Seymour Papert noticed that "in order to ensure a real acquisition of knowledge and to enhance motivation providing a positive feedback, the design of the environment and context should be meaningful for the learners and close to their significant life experiences and circumstances" [16].

Outreach Programs. Although in literature there is not a global unique definition of what K-12 Outreach Programs are, we can define them as the non-formal and informal educational initiatives that schools and universities, as well as companies and associations, promote: they implement or enrich the students' school curriculum or the Teachers' Professional Development (TPD).

Outreach Programs promote a different point of view and methodology, or a new technology that impacts on learners' future choices towards Computer Science (CS) or STEM disciplines. The pedagogies adopted boost learners'

self-directed learning, creativity, collaboration and peer mentoring while removing social and emotional barriers [1].

Sometimes we can find OPs in national or regional initiatives, designed by governments through their educational agencies, that massively address a whole segment of school population in a country, for example primary or secondary schools [14, 8].

Programming is a very popular topic in OP, as it motivates kids to participate in STEM (Science, Technology, Engineering, and Mathematics) disciplines [21, 12, 10, 23] and, having Computational Thinking and Creative Learning as its theoretical framework [5, 22], it can also foster students in persist and pursuing a CS career during their transition from school to University [6].

Outreach Programs attenuate formal roles or hierarchies between teachers and students, and create an environment where kids can hold and drive their own knowledge with a more intentional approach to learning rather than teacher-centered. Hence, OPs are not directly based on an institutional curriculum or framework and do not usually entail the traditional forms of assessment, opting for self-assessment, error resolution, community discussion.

Outreach Programs are definitely complementary to the formal institution, they promote a constant experimentation of tools, environments, scopes and methodologies [4], and tune in with the instances of social construction of knowledge of our era.

The design process. The design in K-12 Computing Education OPs takes different shapes depending on the context in which it occurs. Planning, identifying problems and their solutions, evaluating the outcomes, they all represent aspects of a discipline that support the design and implementation of educational programs.

Researchers, educators or practitioners who concept and plan these pathways do face multiple challenges and devise a few solutions. Sometimes instead, they see the encountered difficulties only after the realization of the program, during the follow-up. For these reasons, our research focuses on the design concerns that K-12 Computing Education programs can occur to OP designers during the ideation, or even happen to them after the implementation of a session.

Based on the literature evidence as well as on additional insights from learning and social sciences, I elaborated the following research question (RQ) in order to model the process:

- **RQ1** - What are the OP design process categories and features?
- **RQ2** - What are the OP design process steps?

The planning of educational paths grounds on a few pillars that guide designers to the choice of an educational or social feature, a type of pedagogical approach, the selection of a target and the possible areas of personal development to be increased. From the analysis of the literature, we highlight these pillars:

1. Principles
    - Skills
    - Self
    - Interaction
    - Citizenship
    - Openness
2. Aims
    - Skills - Increase creativity, critical thinking, positive attitude, persistence, performance, self-regulation; promote STEM disciplines; boost TPD (Teachers Professional Development);
    - Self - Fill the gaps; promote inclusion, equity, engagement, empowerment, self-efficacy and regulation; adopt accessibility and universal design; overcome underrepresentation, bias, imposter syndrome, cognitive load, procrastination;
    - Interaction - Promote social interaction and collaborative learning; increase collaboration, team building;
    - Citizenship - Promote digital awareness and the safe use of the internet;

- Openness - Promote generalization of the programs, scale, reproducibility; adopt iteration, reuse of resources and practices, accessibility.

3. Strategies

- Pedagogy
- TLE (Tools, Languages, Environments)
- Mentoring
- Team/Community building
- Artifacts
- Co-design/customization
- Role-models
- Evaluation/lessons learned
- Dissemination
- Duration

In response to RQ1 I adopted a taxonomy along three assets, which incorporate the different questions to consider while designing a program: concerns influencing the participation of specific categories of learners and boosting their access to computing; practical instances to solve in view of the implementation of a program; design phase and elaboration of the initiative. Some of them set real constraints, somehow impossible to overcome; some are just flexible facets that can be adapted to the goal or targets addressed. On one hand, we have practical requirements to consider and that set a concrete group of items; on the other we attempt to define some general and more theoretical design concepts to exploit during the planning.

In response to RQ2, I delineated the steps of the design process which take into account the above pillars, adapting them in the different educational contexts:

- Goals - Identifying the educational needs and the target/audience, the pedagogy and approach, surveying the state-of-the-art;
- Setting - TLEs adopted, human resources recruitement, location/environment arrangement;
- Motivation - Removing obstacles (emotional, social, infrastructural, logistics...), role-models promotion;
- Teaching/Learning - Mentoring, peer-mentoring, collaboration, tinkering, self-regulation, self-evaluation;
- Artifact - Physical or virtual objects, showcase and dissemination;
- Feedbacks - Pre-session or post-session harvesting, improvement of OP design, accessibility evaluation, co-design;
- Evaluation - Analysis of the impact on learning and interactions, use of the feedbacks;
- Dissemination - Sharing of results (publications, social media campaigns, learners' artifacts anthology), valuation of scale and reproducibility;

In the design process it is necessary to carefully analyse and match the purpose of the initiatives to the addressed target or audience, and also valuing the availability of physical venues and tools. It is crucial to intercept emerging formative needs and real motivation of participants towards computing and technology, but also to guarantee a proper organization and setting to perform the programs.The process is summirezed in Figure 1.

The Outreach Toolkit. Defining the major design concerns will generate a sort of taxonomy of designers' plans during the elaboration of a program. The most common matters can usually be faced on-site with temporary wisdom solutions, but it is also crucial to suggest practical answers to adopt in advance during the planning phase, which can support the effectiveness of the initiatives.

The model I provide in my thesis is not intended to be definitive or inclusive of every aspect of the design, but instead a step toward the scientific community to enhance a reciprocal exchange and collaboration. Because of the breadth of our purpose, I have confidence and will appreciate if my analysis could generate further questions and answers.

The following research questions (RQs) aim to determine the importance of a design toolkit:

- **RQ1** - What are the major design concerns in OPs?

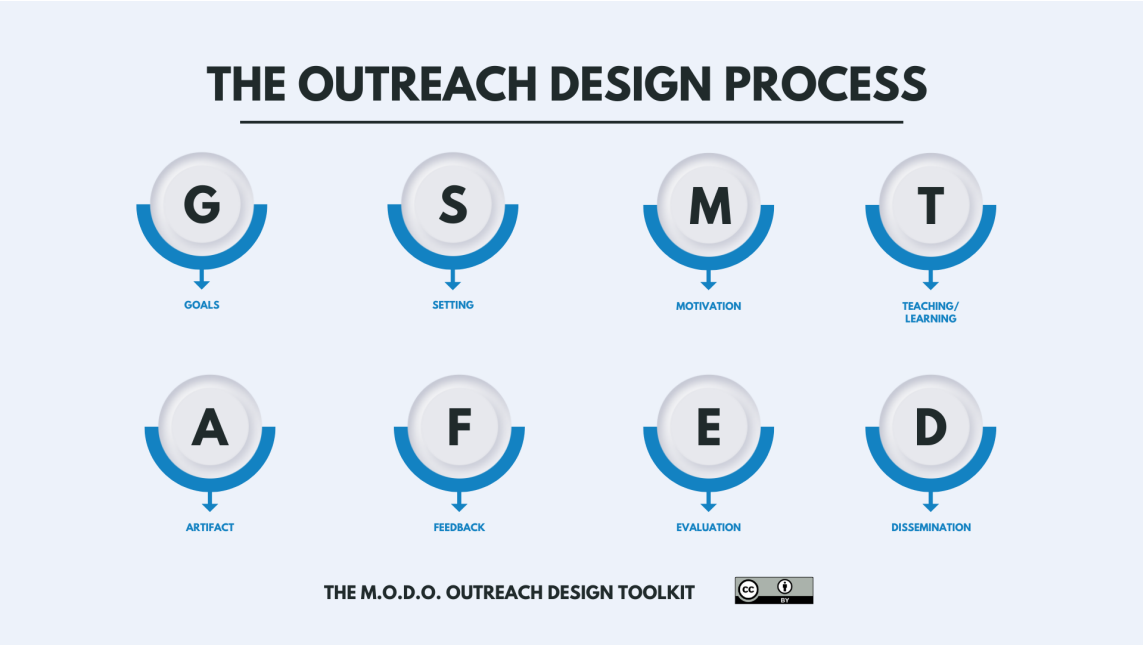

Figure 1: *The Outreach Design Process.*

- **RQ2** - What are the featured criteria on which drawing the guidelines of a toolkit for the Outreach Programs designers?

In response to my research questions, I surveyed the major concern for every above pillar, and later I outlined an OP taxonomy along four criteria, which incorporate the different areas of interest to consider while designing a program, or re-designing after the implementation and follow-up stage:

- **Motivation** - *what can inspire.* It includes the incentives envisioned by designers that can influence or enhance the participation of specific categories of learners, boosting their access to computing literacy. It is also the set of personal expectations, desires and perceptions that students have with respect to STEM disciplines and Computer Science in particular;
- **Organization** - *what is performed in practice.* It includes the operational instances to settle the educational and social environment in view of the implementation of a program;
- **Design** - *what represents the educational purpose and the design process.* It concerns the planning, the elaboration phase and the solutions adopted for the initiative, with regard both to the educational features and the computational ones;
- **Outcome** - *what is the result and can support redesign.* It concerns the forms and quality of the results, they are tangible or not, of the programs and support the redesign stage.

The four criteria features descend from the OP pillars and the major of them are summarised in Figure 2.

A problem still difficult to manage in outreach and not included in the toolkit is the evaluation of the learning outcomes. In fact, it is hard to assess on scientific grounds their effectiveness and impact due to their sporadic nature (duration constraints) and because the methods of implementation and fruition are extremely variable. Regarding the evaluation and possible certification or validation of the outcomes, by the way, there are some interesting institutional frameworks [18] and scientific contributions that can inspire designers [6, 11].

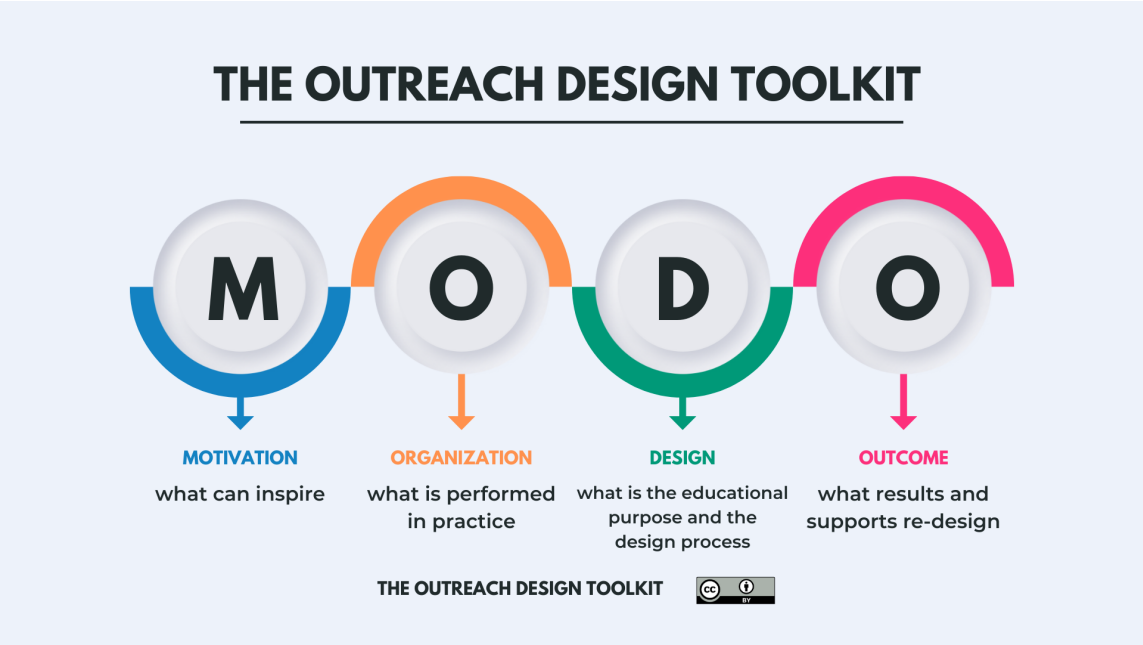

Figure 2: *The Outreach in Computing Education Design Toolkit.*

Potential applications. The toolkit mainly aims to support researchers, educators and practitioners in planning the Outreach Programs. For example, many courses take place in contexts such as summer camps or coding clubs, where volunteers happen to be engaged in solving more strictly pedagogical problems.

The toolkit can support both the design and the follow-up phase for the resolution of the most common and frequent problems, providing guidelines in the single project areas highlighted with the above criteria.

Future directions. One possible development of the design toolkit could concern the application in Teachers Professional Development (TPD), in which relations and self-regulation weigh on the results. In this case, the adults' difficulty in getting involved and sharing one's learning path without hesitation, essentially reversing one's role from teacher to learner, has a considerable effect. Another important future work that might be taken into account and that could be worth investigating is the assessment of the OPs impact, above all by identifying a qualitatively effective system to implement it. Finally, a useful contribution that can represent a future direction to this research work is a more comprehensive definition of a taxonomy of TLEs. In our research lab we are already working on the further implementation of our collaborative Computing Education OP taxonomy platform [3], open to the contribution of worldwide researchers.

Conclusion. Computing Education is a relatively young and rapidly developing field which addresses the demand of technology education and the construction of computing knowledge and competences. It responds in an inter- and cross-disciplinary way, with researchers coming both from the perimeter of Computer Science and Pedagogy. The inevitable and necessary contamination between the two groups generates a few increasingly crucial thoughts both on the theoretical and epistemological level and on the practice of the experiments implemented.

Outreach Programs in K-12 Computing Education are a consolidated reality that is constantly being carried out besides the formal curriculum, and having a formative and social impact on participants. They are the learning environments which integrate elements of the

CS school curriculum with the motivation and social needs of the participants and that base their action on the incentive to participate, collaborate, experiment with languages and tools, filling gaps and digital divide.

Researchers often have to deal with some design concerns and face some specific challenges during the implementation of the programs. The design process, which is described in its stages, in this dissertation is also enriched by a proposal of an operating toolkit which has not yet been found in the literature. The suggested solutions meet most of the design concerns, but we have also highlighted some instructional constraints that are not resolved at the near moment, such as the impact assessment of the initiatives. It would be interesting, as a future direction, to foresee a validation of the educational framework and toolkit with a series of experiments aimed at solving the highlighted concerns.

The toolkit may also have an utility for the practitioners not directly enrolled in research or education, such as the volunteers or company workers who carry out informal outreach sessions. In many countries where there is no CS curriculum, their camps and clubs actually offer the only meaningful access to Computing Education for many students.

References

[1] Agnese Addone, Renato De Donato, Giuseppina Palmieri, Maria Angela Pellegrino, Andrea Petta, Vittorio Scarano, and Luigi Serra. *Novelette*, a usable visual storytelling digital learning environment. *IEEE Access*, 9:168850–168868, 2021.

[2] Agnese Addone and Vittorio Scarano. Outreach in K-12 programming: A systematic literature review on audience and purpose. In *Proceedings of the 14th International Conference on Computer Supported Education, CSEDU*, volume 2, pages 276–283. SCITEPRESS, 2022.

[3] Agnese Addone, Vittorio Scarano, and Carmine Spagnuolo. Towards a collaborative taxonomy of tools, languages and environments in K-12 computing education. In *IEEE Frontiers in Education Conference (FIE)*, pages 1–5, 2022.

[4] Andrew Begel and Amy J. Ko. *Learning Outside the Classroom*, page 749–772. The Cambridge Handbook of Computing Education Research. Cambridge University Press, 2019.

[5] Karen Brennan. A case for why: Society, school, self. In *Computational thinking education in K–12: Artificial intelligence literacy and physical computing*. The MIT Press, 2022.

[6] Adrienne Decker and Monica M. McGill. Pre-college computing outreach research: Towards improving the practice. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '17, page 153–158, New York, NY, USA, 2017. Association for Computing Machinery.

[7] Sally A Fincher and Anthony V Robins. *The Cambridge handbook of computing education research*. Cambridge University Press, 2019.

[8] Adam Fletcher, Raina Mason, and Graham Cooper. Helping students get it: Investigating the longitudinal impacts of IT school outreach in australia. In *Proceedings of the 23rd Australasian Computing Education Conference*, ACE '21, page 115–124. Association for Computing Machinery, 2021.

[9] Tracy Gardner, Hayley C. Leonard, Jane Waite, and Sue Sentance. What do we know about Computing Education for K-12 in non-formal settings? A Systematic Literature Review of recent research. In *Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1*, ICER '22, page 264–281. Association for Computing Machinery, 2022.

[10] Yasmin B Kafai and Quinn Burke. *Connected code: Why children need to learn programming*. MIT Press, 2014.

[11] Jay Lemke, Robert Lecusay, Michael Cole, and Vera Michalchik. *Documenting and assessing learning in informal and media-rich environments*. The MIT Press, 2015.

[12] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. Introductory programming: A systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE Companion, page 55–106. Association for Computing Machinery, 2018.

[13] Victoria J Marsick and Karen E Watkins. Informal and incidental learning. *New directions for adult and continuing education*, 2001(89):25–34, 2001.

[14] Karen Nolan, Roisin Faherty, Keith Quille, Brett A. Becker, and Susan Bergin. Developing an inclusive K-12 outreach model. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '20, page 145–151. Association for Computing Machinery, 2020.

[15] Alannah Oleson, Brett Wortzman, and Amy J. Ko. On the role of design in K-12 computing education. *ACM Trans. Comput. Educ.*, 21(1), dec 2021.

[16] Seymour A Papert. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, New York, 1980.

[17] Joslenne Pena. Seeding the computational skills of diverse non-programmers through non-formal workshops. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*, ICER '19, page 347–348, New York, NY, USA, 2019. Association for Computing Machinery.

[18] Madhu Singh. UNESCO guidelines for the recognition, validation and accreditation of the outcomes of non-formal and informal learning. *PLA Inside Out: An International Journal on Theory, Research and Practice in Prior Learning Assessment*, 1(2), 2012.

[19] Cynthia Solomon, Brian Harvey, Ken Kahn, Henry Lieberman, Mark L. Miller, Margaret Minsky, Artemis Papert, and Brian Silverman. History of LOGO. *Proc. ACM Program. Lang.*, 4(HOPL), 2020.

[20] Claudia Szabo, Nickolas Falkner, Andrew Petersen, Heather Bort, Cornelia Connolly, Kathryn Cunningham, Peter Donaldson, Arto Hellas, James Robinson, and Judy Sheard. A periodic table of computing education learning theories. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '19, page 269–270. Association for Computing Machinery, 2019.

[21] Claudia Szabo, Judy Sheard, Andrew Luxton-Reilly, Simon, Brett A. Becker, and Linda Ott. Fifteen years of introductory programming in schools: A global overview of K-12 initiatives. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, Koli Calling '19. Association for Computing Machinery, 2019.

[22] David Weintrop, Elham Beheshti, Michael Horn, Kai Orton, Kemi Jona, Laura Trouille, and Uri Wilensky. Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology*, 25(1):127–147, 2016.

[23] David Weintrop and Uri Wilensky. To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th international conference on interaction design and children*, pages 199–208, 2015.