



## Colombian Elementary Students' Performance and Perceptions of Computing Learning Activities with Scratch

**Mr. Camilo Vieira, Eafit**

Master of Engineering from Universidad Eafit. Doctoral student in Computer and Information Technology at Purdue University.

Research interests include Computing Education, Computational Thinking and Educational Technologies.

**Dr. Alejandra J. Magana, Purdue University, West Lafayette**

is an Assistant Professor at the Department of Computer and Information Technology at Purdue University West Lafayette. Magana's research interests are centered on the integration of cyberinfrastructure, computation, and computational tools and methods to: (a) leverage the understanding of complex phenomena in science and engineering and (b) support scientific inquiry learning and innovation. Specific efforts focus on studying cyberinfrastructure affordances and identifying how to incorporate advances from the learning sciences into authoring curriculum, assessment, and learning materials to appropriately support learning processes.

# Colombian Elementary Students' Performance and Perceptions of Computing Learning Activities with Scratch

## Abstract

In this paper we present a case study of 117 Colombian elementary grade students' performance and perceptions of a learning activity aiming to promote computational thinking guided by the College Board's CS Principles and Scratch. The lesson plan was designed by the teacher as part of a three-day teacher professional development workshop within an advanced topics course for a master degree in engineering. As part of the workshop, participants were invited to implement their designs in their own classrooms and, together with the researchers, conduct classroom action research. Workshop participants designed their own instruments and gathered data on students' perceptions of the learning module and identified the level of achievement of the selected learning objective. Our research questions are: (a) what are students' levels of achievement of the identified CS principle learning objective as evidenced by their performance on the design learning activity? And (b) what are students' perceptions and engagement with the design learning activity? Student grades, the perceived usefulness, the appeal of the learning activity, and Scratch were considered as positive. Similarly, as part of an exit interview, the teacher participant highlighted the usefulness of Scratch software to implement the learning task guided by the CS principles and backwards design approaches. These results pose significant implications to integrate computing principles and procedures sooner and often into the K-6 curriculum.

## Keywords

Computing science, education, lesson plan, backwards design

## Introduction

Advances in computing contribute to science and engineering discovery, innovation, and education by facilitating representations, processing, storage, analysis, simulation, and visualization of unprecedented amounts of experimental and observational data to address problems that affect health, energy, environment, security, and quality of life. Computing, as both fundamental knowledge and a technical skill, is therefore required to contribute and to compete in our fast-changing and global society<sup>1-3</sup>.

For instance, in the United States, CS10K effort, aiming to have 10,000 high school teachers capable of teaching some AP-level computer science class in 10,000 US high schools by the year 2016, uses teacher professional development programs as the main vehicle for accomplishing this goal. Professional development programs provide opportunities for teachers to bring about change in their classroom practices, attitudes and beliefs, and learning outcomes<sup>5</sup>. Other countries have begun to realize the same need of increasing the number of professionals opting for a computing related career. The US model can be used as a framework for addressing the same issue.

Developing countries, such as Colombia, are not the exception. Colombia faces the possibility of falling behind if they do not keep up with new technological advances<sup>4</sup> and the appropriate training of its future workforce. The Colombian government, as of three decades ago, started significant efforts to improve the informatics and computing knowledge among its people<sup>4, 6</sup>. Alianza Futuro Digital Medellin (AFDM)<sup>7</sup> is an example of these efforts. AFDM is an association between educational institutions and the private sector aiming to transform technician and professional technology education. It consists of eight institutions collaborating to provide the necessary training for high school teachers to be well prepared for enrolling in software-programming courses in higher education.

As part of this effort, high school and technician preparation teachers have been provided with the opportunity to enroll in a master's degree in engineering in the area of educational technologies at Universidad Eafit. As part of this program, teachers participated in a three-day workshop focused on an introduction to computational thinking. Along with this workshop, we conducted an exploratory research study to identify how this group of teachers transferred the knowledge and skills gained in the workshop into their own classrooms.

Here we present the results of a case study of a classroom implementation. The research questions are:

- (a) What are students' levels of achievement of the identified CS Principle learning objective as evidenced by their performance on the designed learning activity?
- (b) What are students' perceptions and engagement with the designed learning activity?

## **Theoretical Framework**

Action research was used as the theoretical framework that guided this investigation. Action research has been defined as a qualitative model of inquiry in which all individuals involved in a specific situation take an active role in the process of the research study<sup>8</sup>. Action research is particularly useful for participants who would like to make changes in their educational activities<sup>9</sup>. We have identified action research as an appropriate theoretical framework to approach our research questions because it has the primary intent to provide a framework to investigate complex working classroom situations<sup>9</sup>.

Action research methods informed this investigation by having the participating teacher formulate a research plan, carry out the intervention and data collection, evaluate the outcomes, and design/re-design further strategies in an iterative fashion<sup>10</sup>. The iterative component is crucial in order to promote a deep understanding of a given situation. This iterative characteristic of action research allows researchers, with the aid of the teachers, to start conceptualizing the problem and move through multiple interventions and evaluations<sup>9</sup>.

For the scope of this paper, we will focus on describing how the teacher designed her educational module and conducted an initial classroom assessment.

## Organizational Framework

The conceptual framework used to make sense of our findings is an educational tool known as Understanding by Design<sup>11, 12</sup>. Understanding by design, also known as the backwards design, is a way of thinking about curricular design and implementation that emphasizes a set of tools and practices that consist of three stages: (a) identifying the desired learning outcomes (the content of the lesson), (b) determining the acceptable evidence of learning (the method of assessing learning), and (c) planning the experiences and instructional approach (or pedagogy). We used understanding by design as the general framework for the workshop because it encompasses all elements that should be involved in any instructional intervention.

## The Workshop

Teachers participated on a three-day workshop where they were exposed to challenges in computing education and respective ongoing strategies to address those challenges in the context of the US and Colombia.

### *Content*

The workshop was designed to provide a general vision of what computational thinking could be, how it could manifest, or where can we find instances. Specifically, the goal of the workshop was to provide (i) an introduction to computational thinking, (ii) an overview of concepts and practices related to computational thinking, and (iii) an exploration of the use of Backwards Design process<sup>10</sup> as a framework to design instructional materials. Some of the key elements of computational thinking discussed were abstraction, problem solving, automatization, complexity, data processing, and the like.

The backwards design process was then introduced<sup>10</sup>. Participants were explained how the backwards design process can be used as a tool to design learning activities. This process was presented together with existing technological environments that can empower the development of computational thinking such as: WISE, Scratch, and Whyville, among others. The major computing topics participants explored were guided by the College Board CS principles<sup>13</sup>. CS principles refer to a framework that describes a set of related learning objectives, computational thinking skills and expected levels of students' performance aiming to increase skills in computing for STEM fields. These CS principles are based on the following computational thinking practices:

- Connecting Computing
- Developing computational artifacts
- Abstracting
- Analyzing problems and artifacts
- Communicating
- Working effectively in teams

They are grouped by seven big ideas and 23 key concepts as shown in Table 1.

**Table 1 – Big Ideas and Key Concepts – CS Principles<sup>13</sup>**

<b>Big Idea</b>	<b>Key Concept</b>
Creativity: Computing is a creative activity.	<ul style="list-style-type: none"> <li>A. Computing fosters the creation of artifacts.</li> <li>B. Computing fosters creative expression</li> <li>C. Programming is a creative process.</li> </ul>
Abstraction: Abstraction reduces information and detail to facilitate focus on relevant concepts.	<ul style="list-style-type: none"> <li>A. A combination of abstractions built upon binary sequences can be used to represent all digital data</li> <li>B. Multiple levels of abstraction are used in computation.</li> <li>C. Models and simulations use abstraction to raise and answer questions.</li> </ul>
Data: Data and information facilitate the creation of knowledge.	<ul style="list-style-type: none"> <li>A. People use computer programs to process information to gain insight and knowledge.</li> <li>B. Computing facilitates exploration and the discovery of connections in information.</li> <li>C. Computational manipulation of information requires consideration of representation, storage, security, and transmission.</li> </ul>
Algorithms: Algorithms are used to develop and express solutions to computational problems.	<ul style="list-style-type: none"> <li>A. An algorithm is a precise sequence of instructions for a process that can be executed by a computer.</li> <li>B. Algorithms are expressed using languages</li> <li>C. Algorithms can solve many, but not all problems.</li> <li>D. Algorithms are evaluated analytically and empirically.</li> </ul>
Programming: Programming enables problem solving, human expression, and creation of knowledge.	<ul style="list-style-type: none"> <li>A. Programs are written to execute algorithms</li> <li>B. Programming is facilitated by appropriate abstractions.</li> <li>C. Programs are developed and used by people.</li> <li>D. Programming uses mathematical and logical concepts.</li> </ul>
Internet: The Internet pervades modern computing.	<ul style="list-style-type: none"> <li>A. The Internet is a network of autonomous systems.</li> <li>B. Characteristics of the Internet and the systems built on it influence their use.</li> <li>C. Cybersecurity is an important concern for the Internet and systems built on it.</li> </ul>
Impact: Computing has global impacts.	<ul style="list-style-type: none"> <li>A. Computing affects communication, interaction, and cognition.</li> <li>B. Computing enables innovation in nearly every field.</li> <li>C. Computing has both beneficial and harmful effects.</li> </ul>

*Assessment*

The workshop participants were asked to design a lesson plan aimed to integrate at least one of the learning outcomes described by the College Board as related to the CS principles. A rubric, described on Table 2, was created and delivered to all the participants to conduct peer-evaluation of all projects developed during the workshop.

**Table 2 – Workshop rubric for peer-assessment**

	<b>1-5</b>	<b>6 - 10</b>	<b>11 - 15</b>	<b>16 - 20</b>
<b>Learning Objectives</b>	The design has no relation with the chosen learning objective.	The design covers the objective partially or its coverage is not clear.	The design covers the selected objective and it is evident	The design completely covers the chosen objective and some others
<b>Assessment</b>	The assessment method for the students is not clear	The assessment method for the students slightly defined. The assessment is teacher-dependent	The assessment method for the students is present and it can be used by and expert, peer and/or self-assessment	The assessment method for the students is clearly defined and is authentic. It can also be used by an expert, peer or self-assessment.
<b>Pedagogy</b>	The selected didactic strategies are not well suited to accomplish the learning objectives	Some of the strategies are appropriated for the learning objectives. Some of the strategies are based on a combination of documented experiences, practices, theory, research and best practices	Most strategies are appropriate for the learning objectives. Most strategies are based on a combination of practical experience, theory, research and documented best practices	All strategies are appropriate for learning outcomes. The strategies are based on a combination of practical experience, theory, research and documented best practices
<b>Reflection</b>	The reflection is not realized or the answers are not coherent with the responses	Reflection is done but it does not provide explanatory elements that allow uncover the experience	The reflection is complete and it allows identify most elements of experience	Reflection is complete and it presents some examples that demonstrate clearly the results, challenges and opportunities of experience
<b>Materials, resources and</b>	The resources and material list is not clear.	Some materials and resources needed to	Most materials and resources needed to	All the materials and resources needed to

<b>technology</b>	The use of technology is out of context	complete the lesson are listed, but they are incomplete.	complete the lesson are listed. The use of technology is relevant and clear.	complete the lesson are listed. The use of technology is relevant and clear. It also offers advantages that make it necessary
<b>Coherence / Order and Presentation</b>	The design is presented in an unorganized way. Every element is isolated from each other	The use of technology is relevant but not necessary The design is presented in an organized way but there are several elements that are not aligned with each other	The design is presented in an organized way and most elements are aligned with each other	All the design elements are aligned with each other to achieve the learning objective in students. All these elements are presented in an organized way

*Pedagogy and Technology*

The three-day workshop combined lecture format with active and cooperative learning pedagogies. The first session consisted of a seminar lecture with one of the speakers in the classroom and the second one via videoconference. Then, during a second session, participants were introduced to the Scratch software and the ScratchEd community.

Scratch is a visual programming language and environment developed by the MIT. It allows creating different kind of media-rich projects including but not limited to: interactive stories, games, simulations, tutorials and music projects<sup>17,18</sup>. Scratch has been transformed, and right now, it has several communities such as <http://scratch.mit.edu> and <http://scratched.mit.edu>. Through these online communities, participants are able to share projects, applications, educational usages, and many other resources.

During the second session started with a tutorial on how to use Scratch and its basic functions and concluded with a set of hands-on activities where participants explored the Scratch environment. During the presentation it was emphasized how Scratch can be coupled with computational thinking and how the ScratchEd community is currently working towards that goal of infusing computational thinking in the K-12 curriculum. Finally, participants working in dyads had two weeks of guided work to develop their own designs. During a third session the participants did a ten-minute presentation of their designs and each of them received feedback from their peer instructors on their implementations.

## Classroom Implementation

This section describes a case study of the implementation of a computational thinking learning module within three groups of fifth grade students. The three groups were exposed to the same learning materials and the lesson was delivered by the same teacher. As mentioned earlier, this teacher participated on the above described workshop where she designed her lesson and received feedback by the workshop instructors and her colleagues.

The lesson consisted of a five-stage process starting with the introduction to the project, which included a description of the expected learning objectives and the competencies to be developed. The lesson then continued with a research stage where students were tasked with identifying the problem to be solved. An analysis and design stages were then carried on where students worked on the potential solutions to the identified problem. The lesson ended with the use of Scratch to implement the problem solution. Finally, students had an opportunity to share their designs and receive feedback from their peers.

### *Content*

The teacher selected the big idea: “Creativity: Computing is a creative activity” from the CS principles for the design of her lesson. Within this big idea, she selected the learning objective number three: “The student can use computing tools and techniques for creative expression.” The learning objective was coupled with the use of Scratch. Therefore, students were expected to be able to use the Scratch computational tool to implement their project.

Specific learning objectives included:

- Use of appropriate Scratch computing tools and techniques for creative expression.
- Use of new forms of expression enabled by computing.
- Select appropriate Scratch computing techniques for creative exploration.

### *Assessment*

Assessment was based on a rubric used by the teacher with three categories: Information Management, Computational Implementation and Creativity Development. Teacher assessment rubric is presented on Table 3.

**Table 3. Teacher assessment rubric**

<b>Category</b>	<b>Top Performance 76-100%</b>	<b>High Performance 51-75%</b>	<b>Basic Performance 26-50%</b>	<b>Low Performance 1-25%</b>
<b>Information Management</b>	Selects, interprets and explains information obtained from the computer	Selects and interprets information obtained from the computer tools (Scratch,	Selects information obtained through computer tools (Scratch, Search Engines and the	Finds difficult to seek and select information via computer tools (Scratch, Search Engines and the



	tools (Scratch, Search Engines and the Wiki)	Search Engines and the Wiki)	Wiki)	Wiki)
<b>Computational Implementation</b>	The Scratch implementation includes movement, appearance and sound controls demonstrating an advanced level of programming knowledge	The Scratch implementation includes some movement, appearance and sound controls which allows identifying some programming knowledge	The Scratch implementation includes some movement, appearance and sound controls but programming knowledge is very basic	The Scratch implementation does not include movement, appearance and sound controls.
<b>Creativity Development</b>	The solution considered environmental variables and constraints. Based on those variables and constraints the student proposes and applies innovative solutions	The proposed solution considered environmental variables and constraints. Based on those variables and constraints the student proposes innovative solutions.	The student identifies environmental variables and constraints for the solution. The student does not propose any innovative solution based on those variables and constraints	The student does not identify any environmental variable or constraint. The solution did not consider any environmental variable or constraint.

*Pedagogy*

Project-based learning and collaborative learning were used as the main pedagogical method. The learning module was presented in five stages: introduction, research, analysis, Scratch implementation, and product socialization. Students working in dyads were given the opportunity to choose the topic of their projects. The five stages were implemented as follows:



**Figure 1. Implementation process**

**Introduction**

During the first stage, the teacher introduced the competencies to be developed, assessment strategies, the learning objective, and supporting tools such as a wiki space and a Scratch tutorial. She then presented the project to the students followed by some examples of how the final product should look like. Dyads and roles were also assigned according to identified student abilities.

## Research

This stage started with an exploratory activity where students identified the influence of information technology in their daily lives. It also included an initial exploration of the Scratch software environment where students had an opportunity to modify existing programs using some basic Scratch controls.

## Analysis

In this stage students worked on the conceptual designs of their projects. Students used a wiki space to describe their initial designs and received preliminary feedback from their teacher. Scratch software was deeply explored together with the teacher. The conceptual designs, roles of team members, and activities to be carried out were also discussed between the teacher and the group members.

## Scratch Implementation

At the beginning of this stage the teacher and dyads of students discussed the expected outcomes of the project, measures of project outcomes, norms and roles for the team, and an additional discussion of difficulties and obstacles students have found up to that point. Then, students were asked to discuss possible improvements or solutions to their obstacles and share them in the wiki space. Students then had two weeks to implement their designs using Scratch. During those two weeks students meet with their teacher two times per week for technical or conceptual consultations.

## Product Socialization

The final Scratch products were uploaded and shared in the wiki space. Projects were then presented to the class and everyone had an opportunity to comment or ask questions to the presenters about their designs. Then, students provided peer feedback on each other's designs including comments of potential improvements, strengths, weaknesses, difficulties and how the technology may have a positive impact for humankind.

## Methods

The purpose of this study was to identify students' level of achievement of the identified CS Principle and associated learning objective as evidenced by their performance on a design learning activity. This study also investigated what are students' perceptions-of an engagement-with the design learning activity. In the following sections we discuss how we approached these two research goals.

### *Participants*

The participants of this study were 117 students from three groups of fifth graders from a Colombian elementary school.

### *Data Collection Method and Data Analysis*

Four instruments were used as the data collection method. One of the instruments consisted of a rubric designed to evaluate the teacher lesson design (see Table 2). This evaluation was performed during the workshop by 16 of her colleagues. The second instrument consisted of a seven-question Likert-scale survey that was used to measure students' perceptions of the learning activity. Survey scoring was based on a five point Likert scale ranging from strongly disagree (1.0) to strongly agree (5.0). This survey was responded by 40 participants in total. Questions from the survey were:

- Learning Scratch is easy
- Scratch is easy to use
- Scratch can be used as a communication tool
- I would like to keep using Scratch
- Scratch helps me to understand the course topics
- I feel confident of using Scratch to achieve the assigned tasks in the course
- I am planning to use Scratch in other projects for other courses.

The third instrument consisted of the score the teacher assigned to the projects based on the rubric described on Table 3. Finally, the fourth instrument was an exit interview with the teacher who expressed her perceptions and possible improvements for following implementations. Descriptive statistics were used to analyze data collected through the teacher rubric assessment, the Likert-scale survey and the student evaluation rubric. Also, a summary of main themes was performed on the teacher exit interview.

### **Results and Interpretation**

The results of the evaluation of the teacher lesson are presented in Table 4. Overall, teacher's colleagues identified her lesson as complete and appropriate. According to rubric in Table 1, the assessment component was clearly defined and well aligned with the learning objective. Pedagogy was also considered appropriate for the chosen learning outcomes. The colleagues' evaluations also suggested that most of the needed materials were clearly identified and that all design elements were coherent and well-aligned with each other to achieve the learning objective in students and all the elements are presented in an organized way.

**Table 4. Results from Rubric Evaluation of Teacher's Lesson Design**

<b>Item</b>	<b>Mean</b>	<b>Normalized Mean</b>	<b>Std. Dev.</b>	<b>Normalized Std. Dev.</b>
Learning Objectives	18.89	0.945	1.76	0.088
Assessment	19.67	0.984	0.71	0.036
Pedagogy	19.11	0.956	1.45	0.073
Reflection	18.33	0.917	2.35	0.118
Materials, resources and technology	18.11	0.906	2.20	0.11
Coherence / Order and Presentation	18.56	0.928	1.74	0.087

The mean score on students' project was of 78% with a standard deviation of 18%. According to the rubric on Table 2, students were able to select, interpret and explain sources of information that helped them take appropriate decisions to inform their designs. Results also suggest that Scratch implementations included an appropriate level of movement, appearance, and sound control demonstrating an advanced level of programming knowledge using Scratch. Finally, the evaluation suggest that students' proposed solutions considered environmental variables and constraints that were then taken into consideration for the design of innovative solutions.

Means and standard deviations of student responses to the survey are described on Table 5. According to these responses it can be suggested that students found that learning with Scratch is easy and it is easy to use. Students also perceived the use of Scratch as a tool to communicate with. Students also found positive responses indicating that they would like to keep using Scratch and that the activity carried out using it helped them understand the course topics. Students also felt confident on their ability of using Scratch to accomplish the goals of the task and that they are planning to continue using it in the future and for other projects.

**Table 5. Usefulness and Usability Survey Results**

<b>Question</b>	<b>Mean</b>	<b>Normalized Mean</b>	<b>Std. Dev.</b>	<b>Normalized Std. Dev.</b>
Learning Scratch is easy	4.53	0.906	0.93	0.186
Scratch is easy to use	4.55	0.91	0.96	0.192
Scratch can be used as a communication tool	4.53	0.906	0.82	0.164
I would like to keep using Scratch	4.43	0.886	0.98	0.196
Scratch helps me to understand the course topics	4.45	0.89	0.85	0.17
I feel confident of using Scratch to achieve the assigned tasks in the course	4.25	0.85	1.10	0.22
I am planning to use Scratch in other projects for other courses.	4.23	0.846	1.19	0.238

During the exit interview, the participating teacher reported very positive experiences on implementing the lesson in her three classrooms. However some implementation problems were also identified. A challenge the teacher faced was related to the timing of the project. She integrated this project toward the end of the semester and students required more time than expected to complete the designs. A second challenge the teacher faced was the amount of guidance students requested from her. What the teacher emphasized the most was the usefulness of Scratch as a tool that allowed her students to conduct their designs. She also pointed out the high level of commitment shown by most of her students who spent more than ordinary class-time to implement their projects. She also pointed out that is her intention continue using Scratch as a pedagogical tool to introduce computing.

## Discussion and Conclusion

The results of this study suggest that the use of Scratch can be used as an appropriate pedagogical tool to integrate principles of computational thinking at the elementary grade level. Results also suggest that overall students had a positive experience using Scratch to implement their designs, and that the learning activity was at the appropriate level. According to the teacher's observations, it seems that the activity was very engaging for the students and that both the teacher and the students demonstrated an interest to keep using this tool.

Although different concepts were discussed during the three-day workshop about what is computational thinking, a good way to establish common ground among participants was through the use of the CS Principles as guidance for teachers to design their lesson plans. It seems that CS Principles served as a leverage or scaffold for these instructors to start working on their designs to then be implemented in their classrooms. Similarly, the use of backwards design process helped teachers to follow a process where they had to jointly consider all elements of design of curricular innovations starting by identifying the learning outcomes (i.e., CS principles), then the appropriate evidence of the learning, and finally the pedagogical approach.

Since computing constitutes a powerful and pervasive tool to solve problems, it has been pointed out that there is the need to start teaching computing early and often in the curriculum<sup>14, 15</sup>. Although this study suggests that computing can be introduced into the curriculum at the elementary level, there are still several challenges that have been identified<sup>16</sup>. Questions such as: "What are the effective ways of learning computing by children?", "What are the effective ways of teaching computing to children?", "What are the elemental concepts of computing?", "How should we integrate the tools [computing machine] with teaching the concepts?" and "How much programming should be required for computational thinking proficiency?" still remain to be investigated<sup>14</sup>.

We believe that this case study provides exploratory evidence that computing activities were well received by Colombian elementary grade students. We also believe that Scratch was an appropriate tool to be used to integrate ideas related to computational thinking. Furthermore, the teacher considered that Scratch software was a useful tool for her teaching and is planning not only keep using it, but is also planning to promote its implementation with other classes from different grade levels and among her colleagues. Also, most of the students indicated their intentions to use Scratch software in the future for other projects. Students also reported a high level of comfort when using Scratch and thought it was easy to learn and useful for their projects.

## References

1. [EOP], "Prepare and Inspire: K-12 Education in Science, Technology, Engineering, and Math (STEM) for America's Future," Washington, D.C.2010.
2. M. Guzdial, "From science to engineering," Communications of the ACM, vol. 54, pp. 37-39, 2011.
3. [NRC], "Report of a workshop on the pedagogical aspects of computational thinking," National Research Council of the National Academies, Washington, D.C.2011.

4. E. J. Cabrales and J. P. Eddy, "Informatics education in Colombia: a study of the present situation and recommendations for the future," *Josurnal of Information Science*, vol. 18, pp. 217-224, 1992.
5. T. R. Guskey, "Professional development and teacher change," *Teachers and Teaching: theory and practice*, vol. 8, pp. 381-391, 2002.
6. M. Palacios, *Por una reforma de la educación técnica y tecnológica para modernizar a Colombia*. 1990
7. EAFIT. (2011, August 20). *Alianza Futuro Digital Medellín*. Available: <http://www.futurodigital.org/>
8. S. Kemmis and R. McTaggart, *The action research reader*: Deakin university press, 1988.
9. W. Hunter, "Action research as a framework for science education research," in *Theoretical frameworks for research in chemistry/science education*, G. Bodner and M. K. Orgill, Eds., ed: Prentice Hall, 2007, pp. 152-171.
10. D. Hopkins, *A teacher's guide to classroom research*. Maidenhead, U.K.: Open university press, 2008.
11. G. Wiggins and J. McTighe, *Understanding by design*, 2 ed. San Francisco, CA.: Pearson Education, 2005.
12. G. Wiggins and J. McTighe, *Understanding by Design*: Alexandria, VA: Association for Supervision and Curriculum Development, 1997.
13. College Board. *CS Principles* - 2011.  
[http://www.collegeboard.com/prod\\_downloads/computerscience/Learning\\_CSPrinciples.pdf](http://www.collegeboard.com/prod_downloads/computerscience/Learning_CSPrinciples.pdf)
14. Fletcher, G. H. L., & Lu, J. J. *Human computing skills: rethinking the K-12 experience*. 2009
15. Lu, J. J., & Fletcher, G. H. L. *Thinking about computational thinking*. 2009
16. Wing, J. M. *Computational thinking and thinking about computing*. *Philosophical Transactions A*, 366(1881), 3717. 2008
17. Scratch. *About Scratch*. Retrieved from [http://info.scratch.mit.edu/About\\_Scratch.2013](http://info.scratch.mit.edu/About_Scratch.2013)
18. Maloney, J., Resnick, M., Rusk, N. Silverman, B. Eastmond, E. *The Scratch Programming Language and Environment*. *ACM Transactions on Computing Education*, vol. 10, No. 4, Article 16. 2010