

Visualizando el Aprendizaje en Resolución de Problemas Abiertos en las Artes

Visualizing Learning in Open-Ended Problem Solving in the Arts

Walter Bender
Sugar Labs. E.E.U.U.
walter@sugarlabs.org

Claudia Urrea
MIT. E.E.U.U.
calla@mit.edu

Resumen

En su artículo, "Hacer el aprendizaje visible" (Urrea y Bender, 2012), los autores describen un marco que hace que los resultados de las iniciativas de reforma de educación a gran escala sean visibles, y comprensibles y aplicables a todos los públicos: administradores educativos, educadores, padres de familia, y los propios niños. En este trabajo, examinamos detalladamente los datos de un concurso de programación entre escuelas del proyecto "Conectándonos", una iniciativa de la computación uno a uno implementado en Costa Rica por la Fundación Quirós Tanzi y el Ministerio de Educación Pública. Aplicamos este marco a los proyectos de Bloques de la Tortuga de 45 niños. Se demuestra una correlación entre la aplicación de nuestro marco al trabajo de los niños y sus resultados de aprendizaje, evaluados subjetivamente. Argumentamos que al demostrar la eficacia de un mecanismo de evaluación de actividades de resolución de problemas abiertos, eliminamos un obstáculo para hacer las artes una parte más dominante de la educación primaria.

Palabras Claves

Construccionismo, Programación, Logo, Bloques de la Tortuga, Diseño y Métodos de Evaluación

Abstract

In their paper, "Making learning visible" (Urrea and Bender, 2012), the authors describe a framework that makes the outcomes of large-scale education reform initiatives visible to and understandable and actionable by all audiences: school administrators, teachers, parents, and the children themselves. In this paper, we examine in detail data from a programming competition among schools from the "Conectandonos" project, a one-to-one computing initiative implemented in Costa Rica by Quirós Tanzi Foundation and the Costa Rican Ministry of Education. We apply the framework to the Turtle Blocks projects of 45 children. We demonstrate a correlation between the application of our framework to the children's work and their learning outcomes as assessed subjectively. We argue that by demonstrating the efficacy of a mechanism for assessing open-ended problem-solving activities, we remove an obstacle from making the arts a more pervasive part of elementary education.

Keywords

Constructionism, Programming, Logo, Turtle Blocks, Design and Evaluation Methods

1 Introduction

The typical use of computing in elementary education is largely bimodal: either children learn to use a computer or they use the computer for learning. We subscribe to the thesis that a netbook computer or tablet can provide children with tools for designing, sharing, and debugging projects that are authentic and, along the way, not only develop the skills to use technology, but also to allow them to design and create fluently with it. Being fluent with technology means being able to express fluently as one would do with a natural language. To design and create things that are meaningful means much more than simply knowing how to

use technological tools since it requires the learner to make decisions about which tool is best suited to create them (Papert and Resnick, 1995).

As children potentially reach this fluency, they are able to design, build, and debug their projects, and at the same time, apply a variety of concepts from different disciplines. During this process, children may develop other higher order skills associated with design (problem solving, modularization, debugging, editing, etc.), computation (sequences, variables, conditions, functions, events, etc.), and learning itself (learning, teaching, reflection, sharing, collaborating, etc.) (Mora et al., 2012). Our goal is to establish a culture of learning and independent thinking in the context of technological fluency. While we currently have no direct measure of how close we have come to accomplishing this goal, we are developing a framework for making learning visible, understandable, and actionable by all audiences: school administrators, teachers, parents, and the children themselves.

1.1 An Argument for Creativity

Creativity is both an innate talent and a skill. On the one hand, some people are born with creative dispositions. On the other hand, any individual may learn to increase his or her creative abilities. A diverse set of skills fuel creative behavior. Creative persons may be open to experience, have a tolerance for ambiguity, an attraction to complexity, the ability to resist premature closure, to accommodate opposites, the ability to sense gaps, a tendency to risk-taking, being self confident, intuitive and with a predisposition to learning (Barron and Harrington, 1981). The creative process is characterized by two distinct modes of thinking: divergence and convergence, which are often used to represent different dimensions of creativity (Parnes, 1988) (Puccio, et al., 2007). Divergent thinking is an expansive mode of thinking. Convergent thinking is a contractive mode of thinking. It is our goal to foster creativity as a skill.

1.2 STEM or STEAM ahead?

It is unrealistic to engage in a discussion of education reform without acknowledging the current emphasis on STEM—science, technology, engineering, and mathematics (Morella, 2012). While we accept the premise that we need more emphasis on the STEM areas, we have a concern that this emphasis comes with a hidden cost: since there is only a finite amount a time available for instruction, adding, e.g., more math, means subtracting, typically, art or music. Some teachers imbue STEM studies within creative problem-solving; often creativity is set aside in deference to achieving curricular goals. We strongly advocate retaining the arts—music and visual arts—as a vehicle for open-ended problem-solving and balancing analytical thinking with both divergent and convergent creative thinking.

We adopt ideas from “Studio Thinking” (Hetland et al., 2007) directly into our use of the computer by emphasizing demonstrations, projects, and critiques—activities that are de rigueur to professional STEM practitioners. Learners are given the platform to develop craft, engage and persist, envision, express, observe, reflect, stretch themselves, explore, and understand. Computation is used as a critical-thinking tool in the context of open-ended exploration and discovery, going beyond the use of the computer as a tool of instruction. Adding an ‘A’ for the arts to STEM results in STEAM, which can power a broadened interest in the STEM curriculum (Bender, 2011) (Maeda, 2011).

1.3 Revisiting Measurement

It is also unrealistic to engage in a discussion of education reform without acknowledging the current emphasis on measurement and evaluation. In this paper, we do not take sides in the debate on high-stake testing, but whereas our goal is to have learning have some positive socio-economic impact on children, we do advocate for an evaluation of our interventions that look more broadly than those data that are captured by standardized tests. We developed a series of recommendations for innovation in evaluation at different levels (Urrea and Bender, 2012): micro (at the level of individual students, teachers, and parents); mezzo (at the level of a classroom or school); and macro (national and global indicators). These mechanisms, briefly reviewed below, are orthogonal to the typical standardized-testing regimes; the two approaches—one serving administrators, the other serving learners—can coexist.

At the micro level, we develop digital portfolios to support reflection that can help students (as well as teachers and parents) be aware of their own learning, and do so by documenting their work and thinking over time. Digital portfolios are part of a “comprehensive system that combines formal, informal, and classroom assessment, including portfolios, to inform the state, the district, the school, and the teacher” (Stefanakis, 2002). Without a way to make visible what students do and what teachers teach, it is difficult to make changes to improve those dynamics.

At a mezzo level, we design tools that help teachers understand the impact and evolution of the program in a larger context—at the level of the classroom or the school. The goal is to design tools that navigate and visualize data automatically derived from the learning activities in which the learners are engaged. These data help teachers, administrators and stakeholders understand the impact of a program and make adjustments to it. The work discussed in this paper is at the mezzo level.

As a macro level, we are developing strategies for understanding the use of computation in learning at a much larger scale. These strategies involve the design and implementation of a repository of objects or artifacts designed by children from different programs. There are a number of similar repositories with artifacts from an individual already in existence, e.g., the Scratch website (scratch, n.d.) and Turtle Art (turtle art, n.d.). Such collections make possible the analysis and understanding of impact at a large scale, and the learning that emerges, not only at the individual, but also at the collective level.

1.4 The Sugar Learning Platform

The Sugar Learning Platform was designed to promote collaborative learning through Activities that encourage critical thinking (Bender et al. 2008) (Bender et al., 2012). Sugar puts an emphasis on divergent thinking. Making that thinking visible to the learner is the goal of our efforts to explicitly introduce assessment tools into the platform and to equally promote cultures of expression and reflection.

Sugar offers an alternative to traditional “office-desktop” software based on the following three affordances: (1) Sharing: Collaboration is a first-order experience. The interface always shows the presence of other learners who are available for collaboration. Sugar allows users to dialog, support, critique, and share ideas with each other. (2) Reflecting: A “journal” records each learner’s activity. It is a built-in space for reflection and assessment of progress. (3) Discovering: Sugar tries to accommodate a wide variety of users with different levels of skill in terms of reading and language and different levels of experience with computing by providing activities with a “low floor” and, where possible, “no ceiling.”

1.4.1 Reflection in the context of Sugar

Sugar supports the notion of “keeping” rather than “saving” one’s work. The interface tries to keep things that offer value automatically in the Sugar journal. The primary function of the journal is as a time-based view of the activities of a learner. As with physical media, such as pen on paper, no explicit “saving” step is needed. The individual journal entries are treated much like pages in a laboratory notebook. There is a title, room for taking notes, and adding tags. The learner is encouraged to adopt a routine whereby time is taken to write about what they are doing either while they are doing it or immediately afterward. This process of note taking becomes the basis upon which they can subsequently engage in reflection (See Figure 1). This mechanism is similar to the “commit message” used in source-code management systems, which would be familiar to software engineers. Sugar journal entries are directly incorporated into digital portfolios, as per the micro level of our assessment framework.

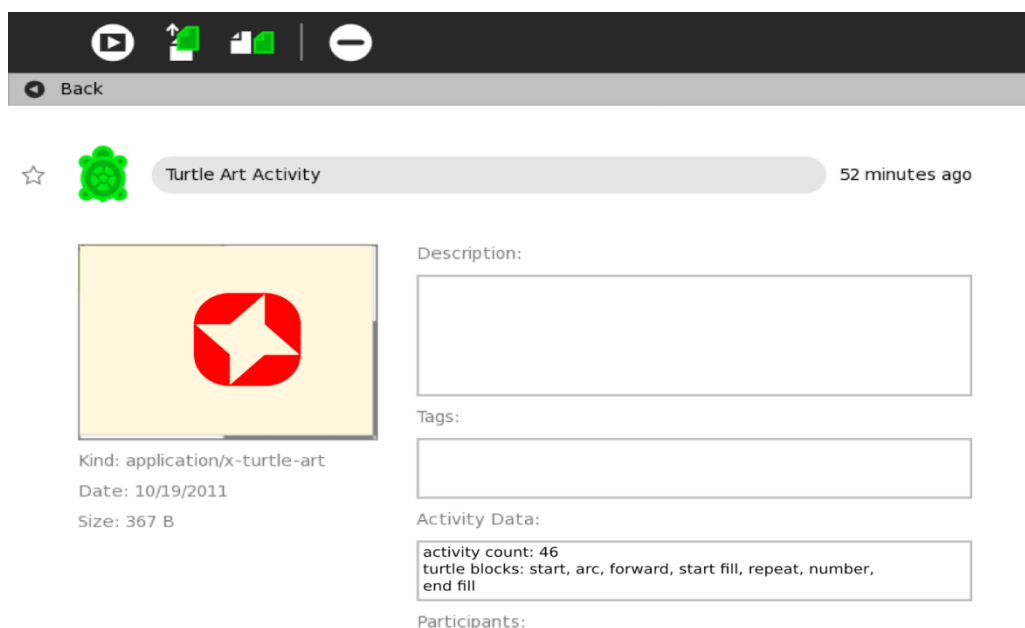


Figure 1: Sugar users can take in-line notes while they are using an Activity. These notes are recorded in a Sugar journal entry.

The Sugar journal has a fixed set of metadata entries that are displayed in the journal detail view for all entries, e.g., “description”, “tags”, “preview”, et al., as well as Activity-specific meta-data. For example, when assessing student work, it is of interest to teachers to know what tools a student may have used and, perhaps how many iterations a student made in creating an artifact. These data may vary from Activity to Activity, hence an enhancement to the journal “expanded view” enables Activities to specify which metadata fields would be useful to display. As shown at the bottom of Figure 1, two fields are displayed: Iterations and Block Types. These fields were set by the Turtle Blocks program (described below). Other Activities may set other fields. This feature enables Sugar Activities to post structured data to the journal that is visible to the student and teacher. They are also of utility for both self and formal assessment as per the rubrics used at the mezzo level.

1.4.2 Rubrics in Sugar

We set out to design rubrics that capture the level of fluency with the technology as well as the creative use of the individual Sugar tools by children. The rubrics associated with the use of the tool captured automatically in some Sugar activities, e.g., Turtle Blocks (See Table 1) and Write. Rubrics associated with the creative process will be assigned manually by evaluators (Urrea and Bender, 2012). The rubrics are similar to test automation frameworks commonly used to verify commercial software systems (Fewster and Graham, 1999), but rather than driving the framework with test data, we run data derived from student projects. From these data we verify that the student has completed a task and compute a “score” for that task. The rubrics for each Sugar Activity are different: they reflect both the nature of the activity and the pedagogical goals of the teachers with whom we developed the rubrics.

In this paper, we examine the automated rubrics generated for the Turtle Blocks Activity as it has been applied to the work of students using Sugar.

1.4.3 The Graph Tool

Whereas one of our goals is to make learning visible to each individual learner, we provide within Sugar a tool for looking at the rubrics for the Turtle Blocks projects that they create (The rubrics are detailed in Section 2.2). The tool analyzes the project data and plots these data (An example is shown in Figure 6 below)¹. At a glance, Turtle Blocks users can get feedback on the types of blocks they used. (The Javascript version of Turtle Blocks, which runs in a web browser, has the graph tool built in to the application, which is also illustrated in Figure 6.)

1.5 Conectándonos educational project

Sugar is the core component of One Laptop per Child’s worldwide effort to provide every child with equal opportunity for a quality education. It is currently used by more than three-million children in more than 40 countries (Bender et al., 2012). Students in Costa Rica used Sugar while participating in the Conectándonos educational program created by the Quirós Tanzi Foundation (QTF) and the Costa Rican Ministry of Education (Fundación Quirós Tanzi, 2012a). Among other activities, the students used the Turtle Blocks programming environment. It is the data from that program that is the subject of our analysis in this paper.

The Conectándonos program was created with the main goals of closing the digital social divide in the country and developing the skills and abilities that its citizens require for a successful life in a knowledge-based society. The organizers of the program believe that the program provides all the necessary conditions for impact: it promotes the use of Sugar; it makes available connectivity and infrastructure at the school and the community; it ensures that each teacher incorporates technology as a learning tool in the classroom, through training and continuous support throughout the year; it involves the community through lectures and educational workshops; and finally, provides technical support to ensure the availability of the tools.

The Conectándonos program benefits more than 4,600 primary school children in 74 schools in different regions in Costa Rica (fundacionqt, n.d.). The Turtle Blocks competition was run in 2012 with the first 15 schools where the program was implemented: five schools in San Isidro de Alajuela, four schools in Río Cuarto de Grecia, three schools in Santa Teresita de

¹ Based upon a Sugar project, Simple Graph, written by a middle-school student in Uruguay who grew up using Sugar and Turtle Blocks in school.

Turrialba, two schools in Curridabat de Curridabat, and one school in San Rafael de la Unión. It is important to note that the three regions with the majority of the schools are located in rural areas.

2. Turtle Blocks

Turtle Blocks (Turtle Blocks, n.d.) is a Sugar Activity with a Logo-inspired (Abelson and diSessa, 1980), (Harvey, 1997) graphical "turtle" that draws colorful art based on snap-together visual programming elements. Programs are created by locking together blocks that represent programming elements. Turtle Block's "low floor" provides an easy entry point for beginners. It also has "high ceiling" programming features that challenge the more adventurous students. As in most Logo environments, in Turtle Blocks, the turtle can exist in three forms: (1) as a robot sharing the same physical space as the child; (2) as a computational object that moves on the screen; and (3) as an abstract mathematical entity.

Turtle Blocks is a "fork" of Turtle Art, written by Brian Silverman (turtle art, n.d.), the author of numerous Logo and block-based programming environments (e.g., Einhorn., 2012 and Lego Mindstorm, n.d.). There are two versions of Turtle Blocks: a Javascript version that is accessed through a web browser and a Python version that runs in the GNU/Linux desktop. The Python version of Turtle Blocks is distributed as part of the core Sugar distribution that is being used by the students participating in the Conectándonos project.

Turtle Art and Turtle Blocks are two members in a large family of block-based programming environments designed for children (Resnick and Silverman, 2005). What distinguishes them from some of its peer environments is its emphasis visual expression. Other environments, such as Scratch put their emphasis on narrative. The Turtle Art focus on art is explicit. As Artemis Papert and Silverman put it (Papert and Silverman, 2011):

Turtle Art is about art. It is a system that is relatively unsophisticated on the technological front and that is quite narrow in terms of the kind of artifacts that can be produced. Turtle Art is focused on creating static images. It is not a general programming environment or a system for exploring math, language, science, etc.

Turtle Art and Turtle Blocks are vehicles to engage children in personal expression. Papert and Silverman argue that by engaging in "deep exploration and produc[ing] substantive works", children become fluent use of technology.

2.1 Using Turtle Blocks

Programming in Turtle Blocks is done by snapping together blocks. Each block is a command for the turtle, e.g., there is a block to tell the turtle to go forward, to turn right, etc. (See Figure 2). The blocks are organized on palettes: one for the turtle, one for the pen, etc. (See Figure 3). Examples of Turtle Blocks projects created by participants in Conectándonos are shown in Figures 4 and 7.

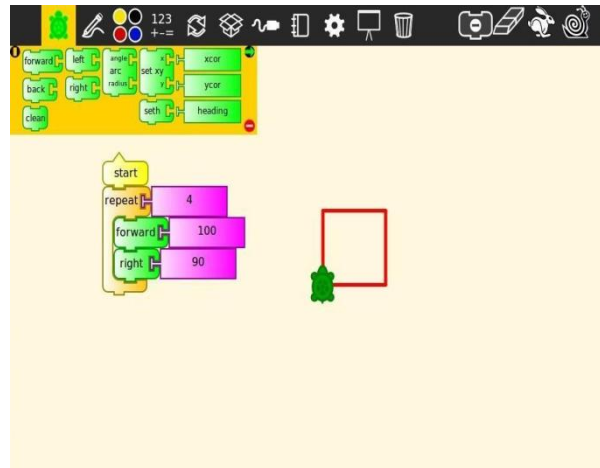


Figure 2: Programs are created by stacking blocks dragged from the block palettes (See Figure 3). Shown above is a program that uses the forward, right, and repeat blocks to draw a square.



Figure 3: Each palette contains a themed collection of blocks that are combined into a program. Left to right from the top are: Turtle, Pen, Number, Flow, Boxes, Sensors, Media, Extras, Presentation, and Trash. Not shown is the Color palette.

For a more detailed introduction on how to use Turtle Blocks see (Turtle Blocks Programming Guide, n.d.).

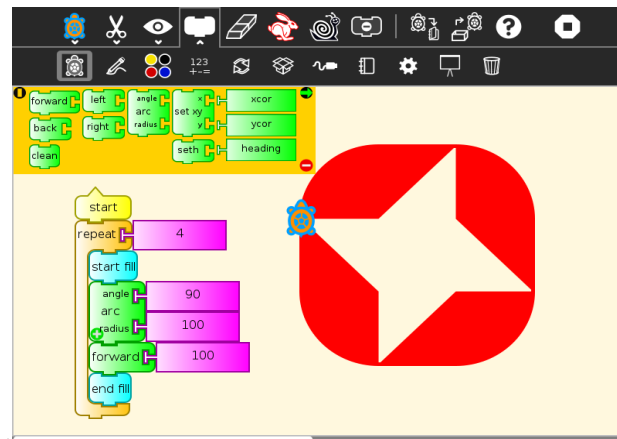


Figure 4: One of the projects from Conectandonos, with the Turtle Blocks program displayed (More examples can be seen in Figure 7).

2.2 Turtle Blocks Rubric

In designing our rubrics, special attention is given to those tools within the Activity that are associated with the main goal of the Activity. For example, Paint is used to create pictures but it also has a text feature. The valuation of the painting tools is larger than the valuation of the peripheral functions. The same criteria apply to Write Activity, used to create a document that may integrate text as well as images and tables. More value is given to the use of tools that allow the user to integrate and format text, than other things such integrate a picture or a table; and to Turtle Blocks Activity, used to program your own art, simulations and games. The tools associated with the Turtle Blocks, Pen and Color, and Flow operators are more important than other tools (See Table 1).

Within the various categories of tools, we apply a further distinction between basic and advanced features. On the turtle palette, blocks for moving the turtle: forward, back, left and right are basic. More advanced blocks, such as setxy, which requires some understanding of Cartesian coordinates, is considered an advanced feature. Similar distinctions are made across the other block categories.

[0, ["start", 2.0], 79, 137, [null, 3]],
[1, "forward", 97, 351, [6, 2, 9]],
[2, ["number", 100], 168, 351, [1, null]],
[3, ["repeat", 84], 79, 183, [0, 4, 5, null]],
[4, ["number", 4.0], 138, 183, [3, null]],
[5, "startfill", 97, 225, [3, 6]],
[6, ["arc", 0], 97, 267, [5, 7, 8, 1]],
[7, ["number", 90], 155, 267, [6, null]],
[8, ["number", 100], 155, 309, [6, null]],
[9, "stopfill", 97, 393, [1, null]],
[-1, ["turtle", "Yertle"], 12.5, 102.5, 0.0, 0, 50, 5]]

Figure 5: The Turtle Blocks project file for the project shown in Figure 4. Both the blocks used in the project and the interconnections between them are listed. It is upon these data that the rubric is automatically applied.

Table 1. Turtle Blocks rubric by type of program block

Total	Category	Details	%	Value Category	Total % per category	Value Detailed	Total % per detail
15.0%	Turtle commands (5%)	Forward, Back, left, right blocks	3.33%				
		Arc, Heading blocks	3.33%				
		Set X and Y coor	3.33%				
10.0%	Pen commands (5%)	Pen up, Pen down, Pen size blocks	2.50%				
		Start fill, End fill blocks, Fill color shade blocks	2.50%				
7.5%	Numerical operators	Basic operations	2.50%				
		Logic blocks	2.50%				
		Random block	2.50%				
2.5%	Coordinates tool	Cartesian or Polar coordinate tools	2.50%				
20.0%	Control blocks (10%)	Repeat, Forever, Wait blocks	2.50%				
		If, If else, While, Stop action, Until blocks	7.50%				
7.5%	Variables	Store in Box1, Box1, Store in Bosx2, Box2	2.50%				
		Box, Store Box Value	5.00%				

7.5%	Procedures	Start, Action1 (definiton and use), Action2 (defintion and use)	2.50%				
		Action blocks (definition and use)	5.00%				
5.0%	Media	Include media objects	5.00%				
5.0%	Extra blocks	Palette of Extra options	5.00%				
5.0%	Sensors	Palette of sensor blocks	5.00%				
15.0%	Creative Process	Programming	15.00%				
100.0%	Total (20%+)		80.00%				
Grand Total			0				

Table 2: The rubric applied to the project shown in Figure 4

Category	%	Details	%	Category Score	Details	Detail Score
Turtle	5	forward, back, right, left	3.33	5	forward	3.33
		arc, set heading	3.33		arc	3.33
		set xy	3.33			
		coordinates	2.5			
Pen	5	pen up, pen down, pen size	2.5	5		
		start fill, end fill, color, shade	2.5		start fill, end fill	2.5
Number		arithmetic operations	2.5			
		logic	2.5			
		random	2.5			
Flow	10	repeat, forever, wait	5	10	repeat	5
		if, while, until	5			
Block		store in, box	7.5			

		action	7.5			
Media		media blocks	5			
Extras		special blocks	5			
Sensor		sensor blocks	5			
Creative process		programming	15			
Sub Totals				20		14.17
					Total	34.2

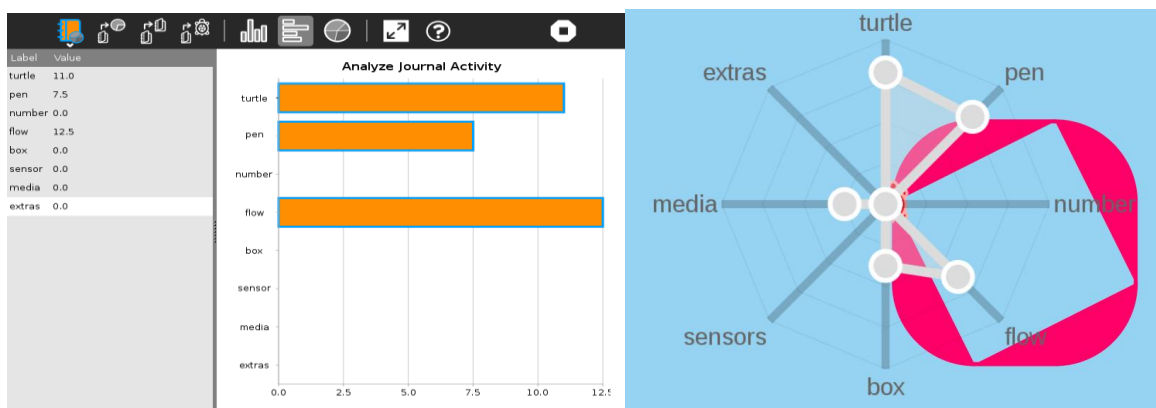


Figure 6: We provide a graph tool (left) with which students can visualize their projects based on the rubric. The analysis shown above corresponds to the project shown in Figure 4. The Javascript graph is of a similar project is shown on the right.

3 Turtle Blocks competition

In 2012, in an effort to promote the use of Turtle Blocks among students and teachers, the Conectandonos program organized a number of activities (Fundación Quirós Tanzi, 2012b): (1) a Presentation and Reflection activity to help teachers get familiar with Turtle Blocks and to encourage them to propose their own projects; (2) an Exploration and Collaboration activity with the Turtle Blocks guide, created by the Foundation. During this activity the teachers explored freely Turtle Blocks and use the guide for reference and support. Teachers worked in teams with other teachers who had different levels of expertise; and (3) a Turtle Blocks Challenges activity to encourage teachers to create using Turtle Blocks any of the figures given to them. At the end of the activity, teachers had the opportunity to present their projects and reflect on the difference between their programs with the original Turtle Blocks program. All of these activities were done with the support of Marco Mendez, a member of Programa de Tecnologías Educativas Avanzadas (PROTEA) from the University of Costa Rica and former apprentice of Artemis Papert and Brian Silverman.

At the end of the year, the foundation decided to organize a Turtle Blocks competition among the children attending their fifteen schools. The contest was open to children at every grade level. Members of the Foundation sent information to the schools, placed posters and visited the classrooms to explain the details of the competition. Some students received time and support from their teachers, and others worked on their projects. The Foundation selected one winning project per school. Due to the large quantity of projects from two schools, two

winning projects were selected in each of those. All the winning students had the opportunity to exhibit their work during the Conectándonos Meeting 2012. More than 150 projects were submitted to the foundation, some of them as a screen shots (a few of which are shown in Figure 7) and others in their original Turtle Blocks data format (see Figure 8). In some cases, more than one project per child was submitted.

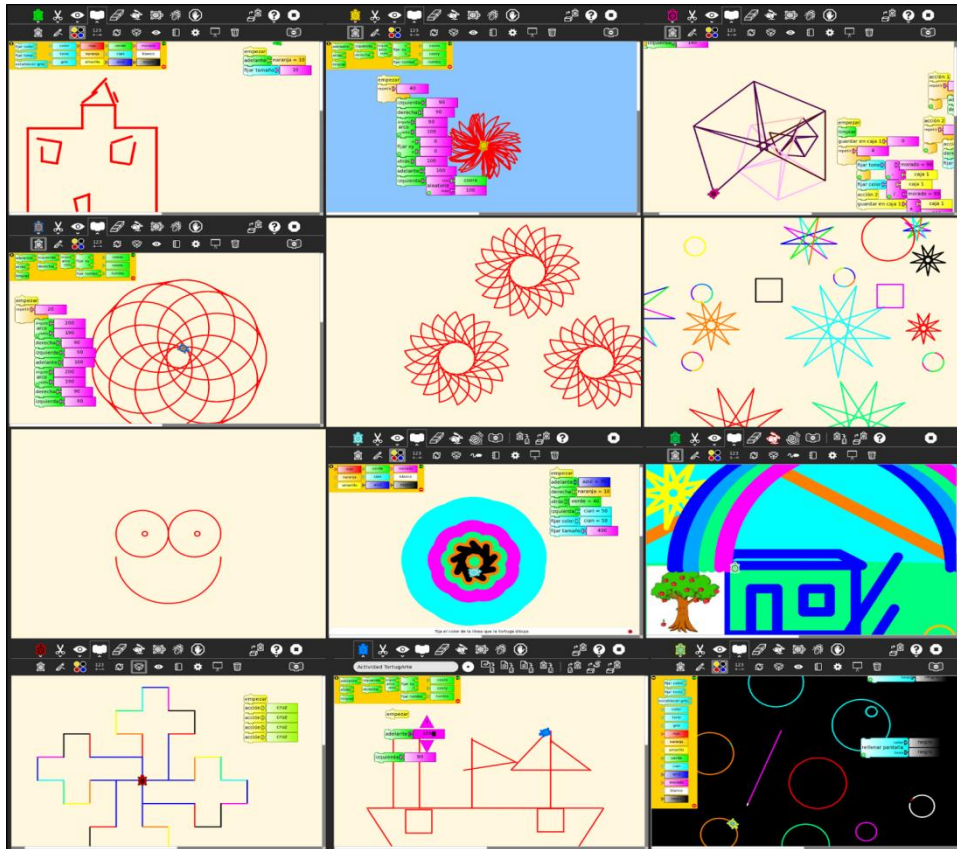


Figure 7: A gallery of some the Turtle Blocks projects created by the children working with QTF.

3.1 Data Analysis

For our analysis, we received 45 project files from QTF (the ones shown in Figure 9). Most of the projects came from school in rural areas; students from schools located in urban areas only submitted five projects. The gender of the child submitting the project was available for only 29 projects: 16 boys and 13 girls. Ages for the winners ranged from 8 to 12 years.

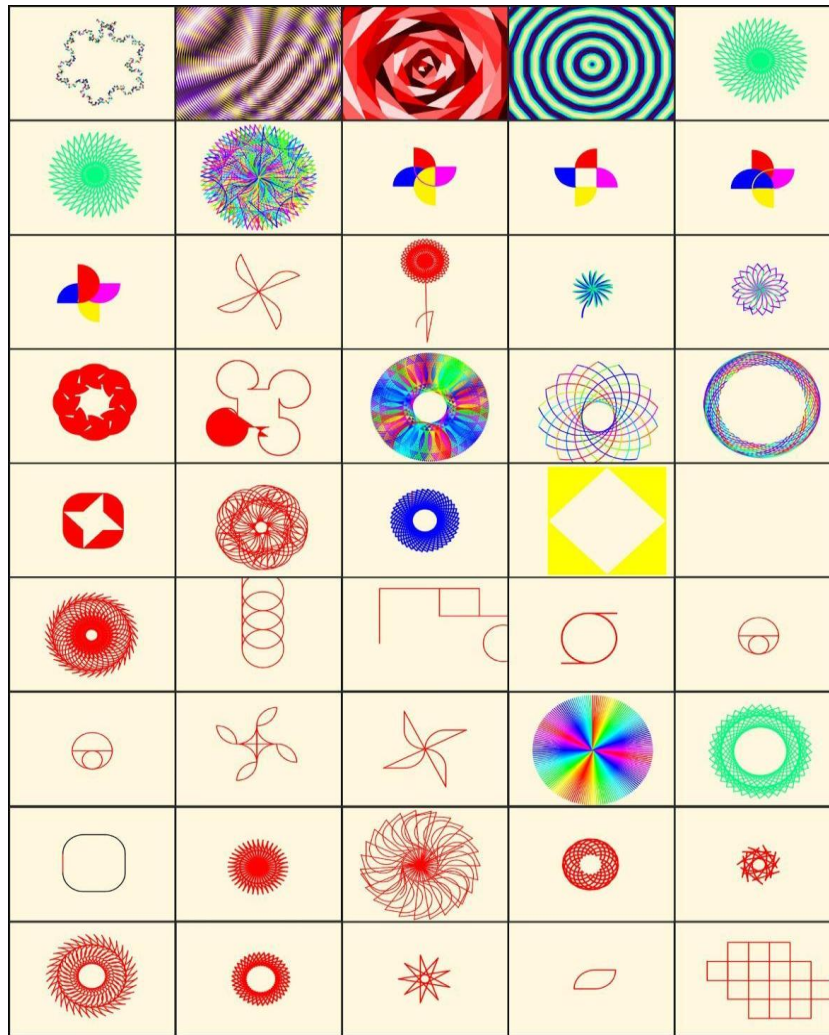


Figure 8: The artwork produced by 45 Turtle Blocks artists. The images are arranged in descending order by their score.

3.2 Observations

The 45 Turtle Blocks projects from QTF were distributed as executable data files, where each file contains a list of blocks used in the project. Also included in the data files is a list of connections between the blocks. These data were used both to recreate the visuals—the “turtle art”—created by the children and to run an analysis of block usage in order to score each project according to the rubric.

A simple automated analysis of all of the blocks found in the data files was run, which tallied scores for each of the categories in the rubric, e.g., turtle blocks, numeric operators, flow, etc. (See Figure 8). Scores ranged from as low as 8.33 to as high as 51.66, suggesting a wide range in project complexity. The mean of the scores is 27.53. The median score is 31.66, indicating that most children were incorporating blocks from multiple categories into their projects (most often, turtle, pen, and flow blocks). Images were also generated for each project, some of which are shown in Figure 8. The scores for the projects were spread across $>4\sigma$ ($\sigma=10.53$).

Rubric applied to Turtle Art projects

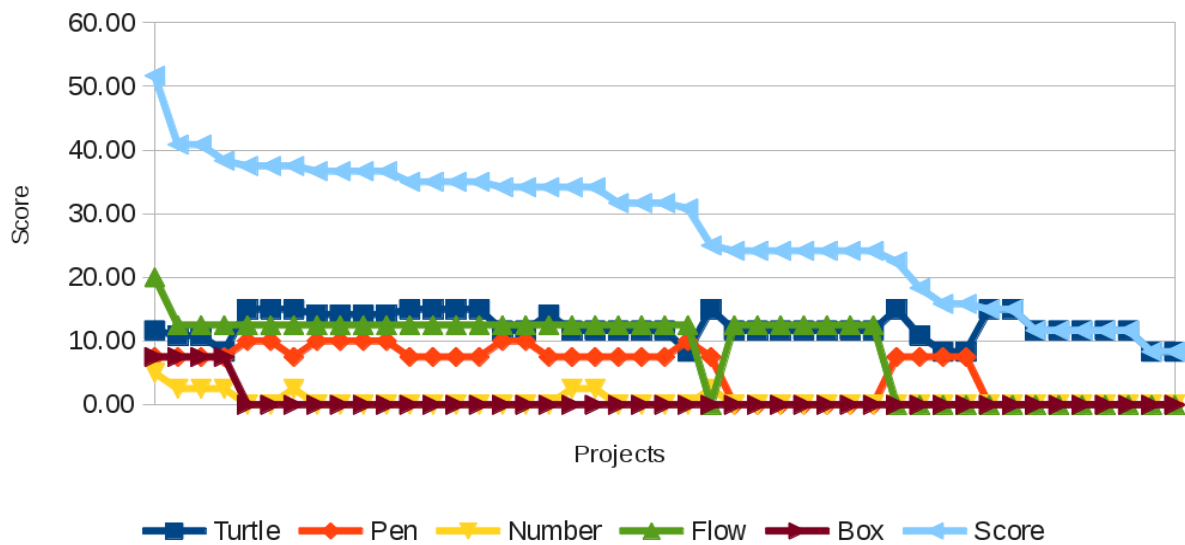


Figure 9: The aggregate rubric scores (light blue) for the Turtle Blocks projects in Figure 8. The variety of blocks contributing to the overall score are shown by the color dots. All of the projects incorporate Turtle blocks; most incorporate pen and flow blocks; while only four projects incorporate boxes (variables).

3.3 Discussion

The images in Figure 8 are sorted by their rubric score in descending order. The first observation is that the visual complexity of the artwork is roughly correlated with the vertical position in the figure, an indication that the rubric is correlated with complexity. The more complex drawings are towards the top and the more simple drawings towards the bottom. Note that the complex rotationally symmetric patterns in the bottom two rows were created by repeatedly hitting the start button, rather than using a repeat block. Had a repeat block been used, these projects would have been scored higher and hence raised higher in the matrix. Also note that there is one blank image. The project file associated with this image had a large number of blocks, but they were arranged such that they did not generate any output.

The following detailed observations are organized by the same block categories found in the Rubric:

- Turtle blocks: It is no surprise that more than 80% of the projects used “forward.” Almost 50% of the projects used “back.” The blocks “right” and “left” were used in more than 90% of the projects. “Arc” was used in 80% of projects, while “setxy,” which is used to move the turtle a specific point in the Cartesian plane, was only used in less than 30% of the projects.
- Pen blocks: Only 40% of the projects used the “setcolor” block to change the pen color from its default, red. None of the projects used the “fill” block to change the background color of the screen. And only five projects (11%) used “setpensize.” Three projects used the “penup” and “pendown” blocks. Eight projects (18%) used “fill” to create filled polygons and arc segments.
- Flow blocks: 32 projects (71%) used “repeat” block. Eleven projects (almost 25%) of the children used a “forever” block; of these six used it for animations. The others

used it as a substitute for “repeat.” In several projects “forever” blocks were embedded inside of “repeat” blocks, suggesting that some children did not really understand what these blocks do.

- Blox blocks: Only four projects used “setbox,” to store a value in a variable. These were among the highest scoring projects and the sophisticated both visually and computationally.

Other observations were not necessarily related to the categories of blocks, but suggest pathways or styles of learning:

- As noted above, a popular theme for Turtle Blocks projects seems to be rotational symmetry. Only five projects did not rely on rotational symmetry, and two of those used translational symmetry. Shaffer discusses using symmetry as an expressive medium in his experiments with young programmers (Shaffer, 1998). Three of the projects were figurative.
- Several groups of projects in Figure 8 were created by the same child. In each case, there is evidence of iteration. For example, although they vary in visual appearance, three of the project in the top of the figure (in the second, third, and fourth positions, counting from the left) were done by the same child. Each is variation from the same code structure, where parameters were modified to produce a diversity of visual impacts. The first, second, and fourth projects in the bottom row were also created by the same child. Since the turtle motion in the graphic that was rotated to create the image in the fourth image resulted in a 180-degree rotation, the result of multiple repetitions did not result in additional visual complexity. The other two images were the result of rotating a graphic at an angle with a higher factor when divided into 360, resulting in a more complex rosette. In several other cases, almost identical graphics were created, but by slight changes to the underlying programs. Other projects, such as Nos. 5 and 6, were duplicates.

3.3.1 Repeat with click

With some of the low-scoring projects, the children used basic turtle and pen blocks to generate a pattern. The pattern, usually a combination of some lines and arcs were not very sophisticated. However, from experience, we have seen that from these simple structures, children create elaborate patterns of rotational symmetry. They do this by repeated execution of their program, taking advantage of the fact that the turtle position and heading are modified with each run. So while some children used the repeat block from the flow palette to generate their artwork, others were able to achieve similar results by repeated manual execution of their programs.

3.3.2 Animations

Several other projects took full advantage of the flow blocks in order to create animated images. Using a forever block instead of a repeat block, a dynamic progression of color and shape was created. Neither the rubric nor the still images capture the intention of the child. Only by running the project is the learning fully visible.

3.3.3 Remixes

About 25% of the projects appear to be remixes (For example, the sequence of four “pinwheels” beginning in the second row of Figure 8).

Creating images through programs has some interesting benefits. One obvious one is that the turtle can draw millions of strokes in seconds. A less obvious one is that it

allows us to collaborate in a way that allows each of us to contribute to the same image. Although this is not unusual for programming it is less frequent in art. Also programming makes it easy and natural for images to “evolve” through a series of variants. Sometimes variant is just a minor change from the previous image. Other times a variant is the beginning of a whole new series (Papert et al., 2010).

As with many programming environments—Scratch, Etoys, etc.—Turtle Blocks comes with many example projects. There are also several portals where projects can be shared (e.g., <http://turtle.sugarlabs.org>, <http://www.turtleart.org>, <http://wiki.sugarlabs.org/Activities/TurtleArt>, <http://www.turtleartsite.appspot.com>, and Facebook). We encourage children to use these examples as starting points to creating their own projects. This has the advantage that children can see blocks *in situ*.

4 Conclusions

We were unable to obtain the subjective evaluations from the teachers regarding the Turtle Blocks projects submitted to the contest. Had we been given access to those evaluations, we would be able to make an analysis of the correlation between those data and our rubric. We do have a list of the winning projects for the competition; only two of those projects were among the 45 projects for which we had data (Nos. 7 and 34). One scored above 1σ of the mean of our metric. The other was one the manually repeated images, and consequently ranked below the mean. Even with additional subjective data from the teachers/evaluators, we can only claim that the rubric serves as a partial evaluation tool for open-ended projects. Partial, because it is still only a measure of how the children used Turtle Blocks to express themselves, but not what they made or why they made it. Some obvious deficiencies in the rubric are its inability to automatically capture the intentions of the students who relied upon “repeat with click”, made animations, or made remixes. Some more sophisticated analysis of the Turtle Blocks program structure may help. But the current rubric does give some assistance to the teacher who is working within the context of accountability, without adding an additional burden of analysis above and beyond looking at the work itself.

Richard Marshall observed that "reliability and validity tend to work in inverse proportion to each other: the greater the reliability of the test, there tends to be a loss of validity, and vice versa." (Marshall, 2009) We do not expect that the rubrics we are using will be in and of themselves a credible measure of creativity, but we do think that they do correlate with the degree of fluency the students have with the use of the tool as an instrument of expression.

With Sugar, we have tried to realize the benefits of student participation in the evaluative process. A focus on self-evaluation, revision, and reflection, manifestations of the Sugar journal interface, may encourage the student to engage in the hard work necessary for progress in every field. Andrea Griswold in her work on student assessment (Griswold, 2006) remarked that "Good writers are not magicians; they are hard workers." The same can be said for good programmers, artists, engineers, musicians, etc. And much of their hard work entails self-assessment, reflection, and revision.

We want children not just to learn about the computer, but also to learn with the computer. Providing Activities such as Turtle Blocks that engage them in computational thinking in the context of personal expression is necessary, but not sufficient. Giving them tools for reflection enhance the learning experience. Giving their teachers simple-to-use mechanisms for assessment increase the odds that Activities like Turtle Blocks will find more mainstream acceptance. Making it easier to assess open-ended projects lowers one of the barriers that are preventing more use of the arts in school.

We reiterate our hypothesis that more arts lead to more creative thinking and real-world problem-solving skills, a hypothesis being tested through longitudinal studies. While we currently have no direct measure of how close we have come to accomplishing this goal, we have some encouraging indicators: In Uruguay, which has been running a nation-wide OLPC program for seven years, we are seeing youths—12 and 13 years of age—engaging in software development in support of the program. Ten percent of the “apps” made available to OLPC users were written by these children and in a recent Sugar release (0.104), 50% of patches came from youths. In a study done in Peru by the Inter-American Development Bank (Cristia, et al. 2012), the results revealed that children that are using XO netbooks, both at school and at home, are five-months ahead from their counterparts in development of cognitive skills as measured by Raven's Progressive Matrices.

5 Acknowledgments

Our thanks to the team at the Quirós Tanzi Foundation for their efforts to promote the use of Turtle Blocks as a tool for design and expression, and for making their data available for the study. Special thanks to the teachers and also the students from the Conectándonos project in Costa Rica. Their work keeps us motivated.

We would like to thank Artemis Papert, Brian Silverman and Cynthia Solomon for their contributions to Turtle Blocks and Pacita Pena for helping design the rubric.

Presentación del manuscrito: 29 de julio de 2015

Fecha de aprobación: 23 de agosto de 2015

Fecha de publicación: 15 de septiembre de 2015

Bender, W. y Urrea, C. (2015). Visualizando el Aprendizaje en Resolución de Problemas Abiertos en las Artes. *RED. Revista de Educación a Distancia. Número 46*. 15 de Septiembre de 2015. Consultado el (dd/mm/aa) en <http://www.um.es/ead/red/46>

References

Barron, F., and Harrington, D. M. (1981). Creativity, intelligence, and personality. *Annual Review of Psychology*, 32, 439–476.

Bender, W. (2011) “Sugar Digest 2011-07-25”, Retrieved from <http://walterbender.org/?p=471>

Bender, W., et al. (2008). *The Sugar Manual*, Floss Manuals (See <http://en.flossmanuals.net/Sugar>).

Bender, W., et al. (2012). *Learning to Change the World: The Social Impact of One Laptop per Child*, Palgrave Macmillan.

- Cristia, J., et al. (2012) "Technology and Child Development: Evidence from the One Laptop per Child Program." Working Paper IDB-WP-304. Washington, DC, United States: Inter-American Development Bank.
- Einhorn, S., (2012). "MicroWorlds, Computational Thinking, and 21st Century Learning", *LCSI White Paper*. (See <http://www.microworlds.com/>)
- Fewster, M., Graham, D (1999). *Software Test Automation*. ACM Press/Addison-Wesley
- Fundación Quirós Tanzi (2012a). *Proyecto Conectándonos*. Unpublished document. Ministerio de Educación Pública (<http://fundacionqt.org/conectandonos.htm>)
- Fundación Quirós Tanzi (2012b). Turtle Art training and competition. Unpublished document
- Fundación Quirós Tanzi (n.d.) http://www.fundacionqt.org/donde_estamos.htm
- Griswold, A. (2006) "Assessment Lists: One Solution for Evaluating Student Poetry", *The English Journal* 96, no. 1 pp. 70-75.
- Hetland, L., et al. (2007). *Studio Thinking: The Real benefits of Visual Arts Education*, Teachers College Press.
- Lego Mindstorms (n.d.) Retrieved July 28, 2015 from <http://mindstorms.lego.com>
- Maeda, J (2011) "From STEM to STEAM: Adding art to science", *New Scientist*, Retrieved from <http://www.newscientist.com/blogs/culturelab/2012/08/john-maeda-steam.html> on 8 August 2011
- Marshall, R. (2009) "Epistemic Vagueness And English Assessment: Some Reflections", *English In Education* 43, no. 1 pp. 5-18.
- Mora, A.; Barragan, S.; and Urrea, C. (2012). "The One to One Model, One Laptop per Child, On the Strengthening of Education and Culture". *Association for the Advancement of Computing Education (AACE) (Ed-Media 2012)*.
- Morella, M. (2012). "U.S. News Inducts Five to STEM Leadership Hall of Fame". *U.S. News & World Report*.
- Papert, A. and Silverman, B. (2011). "Art, Literature, and Turtles", *Proceedings of the 5th International Conference on Informatics in Schools: Situation, Evolution and Perspectives*, ISSEP 2011, Bratislava, Slovakia, October 26-29, 2011.
- Papert, A., Silverman, B., and Bontá, P. (2010), "In the beginning was the Turtle", *Proc. of Constructionism 2010*, Comenius University, Bratislava in association with The American University of Paris, Paris.
- Papert, S. and Resnick, M. (1995) Technological fluency and the representation of knowledge, Proposal to the National Science Foundation, Cambridge, MA.

- Papert, S. & Solomon, C. (1971). Twenty things to do with a computer. *Artificial Intelligence Memo No. 248* and *Logo Memo No. 3*. Retrieved 2012-12-21
- Parnes, S. J. (1988). *Visionizing*. East Aurora, NY: D.O.K Publishers.
- Puccio, G. J., Murdock, M. C., and Mance, M. (2007). *Creative leadership: Skills that drive change*. Thousand Oaks, CA: Sage Publications inc.
- Resnick, M., and Silverman, B. (2005). "Some Reflections on Designing Construction Kits for Kids", *Proceedings of Interaction Design and Children conference*, Boulder, CO.
- Shaffer, D. W. (1998). *Expressive mathematics: Learning by design*. Unpublished doctoral dissertation, Massachusetts Institute of Technology, Cambridge, MA. 14 Oct, 1998
- Scratch website (n.d.) Retrieved from <http://scratch.mit.edu/>
- Stefanakis, E. (2002). *Multiple Intelligences and Portfolios: A Window into the learner's Mind*, Greenwood Press.
- Turtle Art (n.d.) Retrieved from <http://turtleart.org>
- Turtle Blocks (n.d.) Retrieved from <http://turtle.sugarlabs.org>
- Turtle Blocks Programming Guide (n.d.) Retrieved from <https://github.com/walterbender/turtleblocksjs/blob/master/guide/README.md>
- Urrea, C. and Bender, W. (2012), "Making Learning Visible", *Mind, Brain, and Education*, 6, pp 227–241. doi: 10.1111/j.1751-228X.2012.01161.x