



HAL
open science

Taking Programming into Kindergartens: Exploratory Research Activities Using ToonTalk

Leonel Morgado, Maria Gabriel Bulas Cruz, Ken Kahn

► **To cite this version:**

Leonel Morgado, Maria Gabriel Bulas Cruz, Ken Kahn. Taking Programming into Kindergartens: Exploratory Research Activities Using ToonTalk. Eurologo'2003 Conference, 2003, Coimbra, Portugal. pp.178-189. hal-00190134

HAL Id: hal-00190134

<https://telearn.hal.science/hal-00190134>

Submitted on 23 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Taking Programming into Kindergartens: Exploratory Research Activities Using ToonTalk

Leonel Morgado, Maria Gabriel Bulas Cruz
Universidade de Trás-os-Montes e Alto Douro (UTAD)
Eng. II, Quinta de Prados, 5000-911 Vila Real, Portugal
leonelm@utad.pt

Ken Kahn
Animated Programs
49 Fay Avenue, San Carlos, CA 94070, USA
KenKahn@ToonTalk.com

Abstract

We believe that computers should be introduced in kindergarten education under the same philosophy and methods employed for other elements of modern society. Children should be familiar with computers in general, not just with specific sets of skills. We see computer programming in particular as a way to empower children, allowing them to grasp fundamental computing concepts, and not fearing to use a computer in novel ways; they should be able to approach the computer-tool as an adaptable tool, without limited purposes.

For the past three years, we conducted exploratory activities, to grasp overall problems and requirements that such research would face. Here we'll present main themes under evaluation and preliminary findings.

We hope that the overall picture composed by this unprocessed data will serve as a basis for launching much-needed future research on this field, distinguishing computer activities among themselves.

Keywords

ToonTalk, concurrent programming, children programming, programming by demonstration, kindergarten, early childhood, constraint programming

1. Introduction

By “kindergarten” we refer to the Portuguese variety, with ages 3 to 5, in the same room (about 24 children), with one teacher. At first sight, the most obvious hurdle towards introduction of computer programming here is the absence of reading and writing skills. Several approaches can overcome this, but one first asks: why should we bother, in the first place? What purpose can programming serve at the kindergarten level?

Kindergarten education aims to enhance children’s basic skills: personal and social development, expression and communication (motor, drama, art and musical expression, oral and written language, mathematics), and knowledge of the world (Ministério da Educação, 1997 and 1999). The key word is “enhance” (we might as well

say “awaken”): the point is empowering children to more easily and fully learn as much as they can from the world that surrounds them and from their own life experiences.

Computers are prevalent in today’s society; but it’s not just a matter of learning basic skills, associated with specific software; to tap the full power of a computer system, one must acquire «*a level of technological fluency (that has to develop over years) equal to the levels of reading fluency we now regard as basic skill*» (Papert, 1998). It is this power that allows users to adapt computers to suit their needs.

Furthermore, research has shown that children, even at the pre-reading level, benefit from “*greater developmental gains when compared to children without computer experiences in similar classrooms*” (Haugland, 2000). Regarding programming experiences at kindergarten, research has been limited, but available results seem to point in the same direction (Degelman et al., 1986; Klein et al., 2000).

However, traditional computer programming requires effort, dedication and persistence, “*like learning a foreign tongue*” (Smith and Cypher, 1999). These authors state that programming should search for ways of getting closer to human communication, making it possible for every human being to benefit from the computer tool’s ability to be adjusted to different needs. The same approach is mentioned in the ACM paper, “Strategic Directions in Human Computer Interaction” (Myers 1996), where end-user programming is one of the strategic research and development areas listed.

Our research was motivated by the coming into existence of a new kind of programming language: ToonTalk (Kahn, 1996) is the first animated programming language. “Animated”, in this context, means that the code itself is animated, as in a cartoon movie (i.e., it is not a simple addition of animated elements to an existing language concept). ToonTalk is also fundamentally different, at the level of the computer science behind it, from other child-oriented programming languages, such as Logo (Logo Foundation, 2000), StageCast Creator, formerly known as KidSim (Smith and Cypher, 1999) or Squeak (Ingalls et al., 1997), because it is a Concurrent Constraint Programming language (Saraswat, 1993). The programming style is described in more detail in section 3, within this paper.

This new approach to programming might yield different overall results. However, how should programming activities be conducted? Perhaps focusing on children’s autonomy and learning achievements? By performing what kind of programming activities?

We believe that different responses to different elements of programming (assignments, evaluations, debugging, etc.) are to be expected. And also that programming activities shouldn’t be performed as a goal in itself: there should be an effort to integrate them in the overall objectives of each kindergarten.

Since it was expectable that we would face several different problems and responses, we have been conducting several exploratory activities since May 2000, in order to build up an experience base for more systematic research.

2. Research Settings

Field activities were initiated in May 2000, upon the release of the European Portuguese version of ToonTalk. Since then, there were 4 major settings for these activities, detailed in Table 1. Theoretical research is underway, and we intend to launch focused research activities within the next few months.

Table 1. Research settings

Setting	Period	Description	Groups of Subjects (Age Gender)
1	May/2000 to Jun/2000	Inside the kindergarten room, two kids at a time, initial trials.	(5M, 4M), (4M, 5M), (4M, 4F, 5F). Total 7
2	Feb/2001 to Jun/2001	Outside the kindergarten room, 6 kids (in pairs) at a time. The pairings would be informal (i.e., the children would form the pairs themselves). The groups with 3-year olds were conducted separately.	(4F 3F), (3F), (5F 4F 5M 5M 5F 5M), (4M 4F 5F 5F 4M 4M), (5M 5M 4M 4F 4F 5M). Total: 21
3	Nov/2001 to Feb/2002	Identical to setting 1, but benefiting from acquired experience. Interrupted due to lack of time to perform the sessions.	(4F 4M), (4F 5M), (4M 4M). Total: 6.
4	Oct/2002 to Mar/2003	6 computer-activities teachers received specific training and are conducting ToonTalk activities in one kindergarten room each, once per week. There is a weekly meeting for coordination and development of new activities.	Data collection is still underway in the kindergartens. Number of children in each: 18, 12, 18, 21, 19, 8. Total (tentative): 96

On all settings, 5-year olds and 4-year olds were not segregated into specific groups, although some instances did occur on setting 4. 3-year olds, however, were never paired with older children: they took part in the activities, but were paired among themselves.

2.1. Setting 1

Due to the short time before summer holidays ensued, the initial trials were set around specific themes that we could use as a basis for the following year. These were detailed in a different paper (Morgado et al., 2001), but can be summarized as:-

- Session durations from 20 to 40 minutes are viable;
- Generalization in programming was achieved, employing a “usefulness” approach;
- Two approaches to teacher intervention were essayed:-
 - Directed: children were suggested activities, and then tried to achieve them, following the teacher’s hints. The teacher would query the children, debate their interpretation of events and help them decide what to try out.
 - Coached: children decided what they wanted to do; only sporadic suggestions were presented. On queries and debating, teacher intervention was identical.
- Activities in the coached sessions were simpler, but children achieved greater autonomous control over the programming environment: moving objects, resizing, erasing, etc.

All children but one were selected by their kindergarten teachers, on the basis of having demonstrated ease of use of computers (drawing, using CD-ROMs, etc.). One child took advantage of a momentary absence of the teacher, on the researcher’s first day at the kindergarten, before the first session, to tell him that “the teacher said that I should be here”. And so, she took part on the following sessions as well.

2.2. Setting 2

Children came in groups of 6 to a University room, set aside for them; they would play in pairs, in 1-hour sessions (actual computer use time would vary, but the usual duration was around 40 minutes). The activities were conducted in mixed groups of ages and genders. Three of the children were 3-year olds, and those were in two separate groups.

With all groups aged 4 and 5, the coached approach from the previous year (setting 1) was used: given that children had enjoyed ToonTalk, we assumed this approach might provide greater insights on their progress and hurdles. For the children aged 3, given that we had no previous experience, the directed approach was used. The general description of these sessions was covered in another paper (Morgado et al., 2002).

At the cooperating kindergartens, a meeting was held with the parents of all enrolled children, and no selection was performed: the children that took part were all whose parents came to the meeting and expressed their willingness to have them participate.

With 3-year olds, major aspects were:-

- Children managed to program an “exchange” procedure (ToonTalk robot) as early as the second session, following hints and suggestions;
- Children made reference to ToonTalk knowledge from previous sessions and used it (e.g.: the notion of “tidying up” things before giving them to robots, or that turning pictures around turned off their behaviour);
- There seems to be a noticeable gap between what 3-year olds can understand and their limited ability to make use of it due to mouse-control woes.

Major aspects of the sessions with 4-year and 5-year olds are presented in section 5.

2.3. Setting 3

These sessions were performed in the fashion of Setting 1, but always using the coached approach. The focus was using the experience from Setting 2 sessions on children without previous ToonTalk experience. Although only a few sessions took place (professional demands on the researcher that was performing the sessions rendered prosecution impossible), they served to further consolidate the acquired experience.

Children were selected by the kindergarten teacher, on the basis of being the ones less prone to miss class (i.e., maximizing the chances that they would be present at the kindergarten at the session schedule). It was emphasized to the teacher that she should not simply pick up the “brightest” children, something to which she agreed.

An interesting fact, however, stands out: some of the children on the same kindergarten had taken part on Setting 2 sessions at the University, the year before; ToonTalk was installed on the kindergarten computer, but its use was scarce, if at all (and without any adult assistance whatsoever), because the teacher did not know how to use ToonTalk. And yet, several children came to the researcher and could correctly explain the sequence of actions used to build a house, as well as mention other aspects of the game. This was a demonstration that children had understood the process, because not only they could reproduce it, but they could explain it to other children that hadn't used ToonTalk before. But it also shows that they had enjoyed the sessions, because there was a fair amount of enthusiasm: children wanted to try out what their colleagues had explained, even though they had not been selected by the kindergarten teacher.

2.4. Setting 4

6 kindergarten computer-activities teachers received ToonTalk training, before setting to use it on one kindergarten room per week each. All children on these rooms took part. In the first session, the teachers let the children get acquainted with the environment. Then three different approaches were suggested:-

1. “Straight into programming”. Or “programming as the basic activity”. At the second or third session the latest, robot-programming activities would be suggested, and other manipulation activities (enlarging, vacuuming, etc.) would only be practiced as necessary.
2. “Automated behaviours”. Or “programming as an extension based on automated behaviours”. Children would explore environments based on cause-effect elements, like joining a truck, a box and a robot to build a house; playing with automated sensors; giving objects to carrier pigeons that carry them to their nests; combining properties and see the resulting behaviour of objects.
3. “Acquaintance first”. Or “programming as a development of generic activities”. Children would gain basic control over the programming environment, enlarging and shrinking objects, copying them, customizing the looks of the environment, and using it as a place for conducting usual kindergarten activities: pattern matching, knowledge of professions, of animals, of cookery, etc. Programming would only be introduced afterwards, integrated in the environment.

As the sessions went on, the reporting process was tuned (explaining to teachers what information was relevant), and weekly meetings have helped on several accounts:-

- Teachers themselves came across some of problems also faced by children (programming without a purpose, controlling tools, etc.);
- Teachers actions tended to shift towards their beliefs (“The 3-year olds are too young, I haven’t used it with them”; “They are too young to program, I just did mouse-control”), persistence and examples were needed to keep them on track;
- Teachers would often see ToonTalk itself as the goal, hands-on examples were required to show them how to fit ToonTalk into the existing kindergarten themes and activities, and maintain research goals.

3. Programming in ToonTalk

ToonTalk is both a Concurrent Constraint Programming Language (Saraswat, 1993) and a programming environment: children play and program in a city (metaphor for the entire computation), flying around in a helicopter with a small figurine or avatar that can get into houses (metaphor for agents, actors, processes or objects), where robots (methods) can be programmed. Programming uses objects such as boxes (tuples, arrays, vectors or messages), scales (comparison tests), trucks (agent spawning), bombs (agent termination), notebooks (program storage), text and number pads (constants), and birds and nests (channel communication). Robots are programmed moving them with the mouse, performing required actions (method actions), inside their thought bubbles. After training, the method preconditions are visible in the thought bubble. These can be relaxed (generalized) by use of a vacuuming tool to erase/vacuum: an erased condition retains only its type (all values are valid); a vacuumed condition is no condition at all.

Any picture can also be thought of as a process or object, since robots can run on its back, performing all sorts of actions. Finally, pictures can be exported from the ToonTalk programming environment, their code being translated into Java, for use on Java-enabled environments such as Web browsers.

For instance: to program a robot to exchange two numbers in an array, a child gives it a two-hole box (the array), with a number on each hole. The robot enters its thought bubble and the child moves it with the mouse, to pick up the first number, drop it, pick up the second one, drop it inside the first hole, pick up the original number and drop it inside the second hole. The robot will only perform the exchange using the numbers that the child used (displayed on the robot's thought bubble). If the child uses the vacuum cleaner to "erase" the numbers from the robot's thought bubble, it acts upon any two numbers. If numbers are vacuumed instead, the same robot will act upon any box contents. Some papers provide further details (Kahn, 1996; Morgado et al., 2001). Figure 1 shows a programmed robot in action, dropping number "2" on another number, which is in fact a remote sensor for the position of a picture. The programmer's hand (cursor) is clearly visible beside the robot.

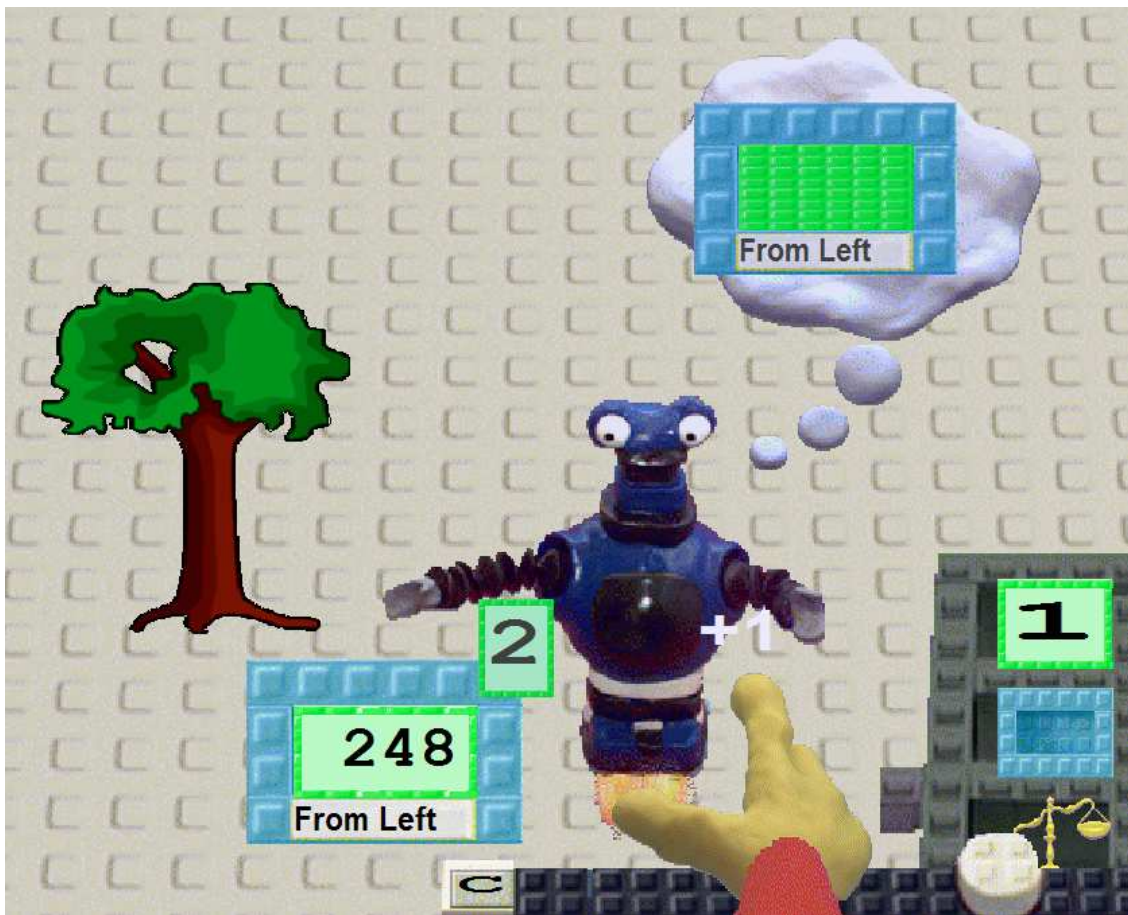


Figure 1. Robot moving tree by adding 2 to its distance from the left.

4. Initial Set of Activities

Under Setting 1, proposed activities were quite simple (Morgado et al., 2001): robots to exchange pictures, erase objects, send pictures using a bird, and assembling two of these one in order to have a pair of ball-passing robots (Morgado et al., 2001). For Settings 2 and 3, activities were mostly suggested by children first, and then adapted by the teacher

to the ToonTalk environment and language. Children built houses, taught robots how to do it for them, customized the city, combined a roof-writing robot with a house-building robot, taught robots how to write their names, etc. Sometimes, children would teach robots with no purpose at all, just for the sake of “playing with the robot”.

5. Conceptual Hurdles for Children while Programming

Under Setting 2, there was only a computer teacher for 6 children at the same time, and this allowed each pair of children more room for failure (less direction). This allowed us to become aware specific skills, or specific moments in which the ability to complete programming activities could be compromised:-

- the initiation of the programming process (giving a box to the robot) may not be understood as an example or starting situation, but as an “interrupt” or “switch”, like turning on a toy, therefore lacking the association between preset conditions and intended action; consequentially, the concept of an activity being taught as something with IN and OUT conditions (or “before” and “after” state), using a box as a container for acting upon, is not absolutely clear: quite often a child will start teaching a robot by giving it an empty box (using it like an “on” switch), and then “teach” with objects taken from the tool box or notebook;
- most children find it hard to grasp the “end of program” concept: after teaching a robot, they often proceed by playing around with the toolbox objects, regardless of the fact that they are moving them with the robot’s hands, not the programmer’s;
- the concept of sequencing activities taught to robots can be confusing: often, children expected robots to start “playing” the moment they finished teaching them, forgetting that they had to give them a starting box; but they wouldn’t make that assumption for robots intended to be used in a specific situation, such as, “when it reaches the new house”, or “when the picture is turned around”;
- “debugging” is an extremely alien concept. Most children assume that a robot learned exactly what they meant to teach him, to the point of sometimes being satisfied with the “teaching” *per se*, not bothering to check if the robot worked; also, they would be puzzled upon seeing a robot repeat the redundant mistakes they made while programming it (like dropping a box on the wrong place and picking it up again), for they would forget about such deviations from the intended path, remembering only the intended path or goal.

At a more environment-specific level, we became aware of issues to address in order to further children’s autonomy with the software; but these hurdles are specific to the ToonTalk environment, not to children’s reasoning. This proved valuable as guidance to computer teachers, on setting 4. Some examples: drawing children’s attention to focus cues (objects wiggle and flash), training tools’ behaviours (enlarging/shrinking, vacuuming/spitting, etc.), having terminology adapted to young children (“drawing” not “image”, “toys” not “objects”, etc.), defining logic explanations for design restrictions (“Why do birds carry only square things?”, “Why do robots demand boxes?”), etc.

6. Lessons from Working with Kindergarten Teachers

The teachers conducting activities in setting 4 were computer-activities teachers for kindergartens (profile: computer training + education degree or specific training). Each

provides computer-activity support at 4 kindergarten rooms per week, works directly with about 100 children and provides computer training for regular teachers based at those kindergartens.

A specific document was laid down for guidance, setting forth the data recording style and elements, explaining the approach styles mentioned in section 2.4, emphasizing the importance of maintaining a given approach, providing examples of activities for each approach style, explaining the issues put forth in section 5, establishing terminology, and providing some ready-made logical explanations for children’s anticipated questions. (Example: “Why must we stop teaching the robot by pressing ESC?” – “Because it’s flying, while in the Robot School in the Clouds. If we don’t tell it to come down, by pressing ESC, all the stuff will fall and on the ground and get broken.”)

Quickly we found out that computer teachers involved in programming were facing some of the hurdles we had previously found with children: the very first time, they taught a robot... with an empty box! (The very first hurdle mentioned on section 5.)

Therefore, a specific strategy was proposed for presenting robot programming:

Table 2. Strategy devised for teaching robot programming

Step	Challenge question	Intended consequence
1	What do you want to do?	Setting a purpose.
2	What do you need?	Define needed objects and assemble them in a box.
3	How do you do it?	Execution of actions upon the objects.
4	Can you teach a robot that?	Demonstrate to robot a specific set of actions.
5	Did it learn properly?	Emphasize the need to test programmed actions.
5a	What went wrong?	If the robot wasn’t properly programmed, or if it doesn’t connect to other objects as expected, discuss the reasons and possible solutions.

The teachers found this set of question helped them see programming in a clearer way, but it’s too soon to evaluate its impact and usefulness with children.

Another problem faced was that teachers found ToonTalk to be so different from what they were used to, that their activities would be closed within ToonTalk, disconnected from everyday kindergarten activities. Sometimes, there would be no sense of purpose for the activities, beyond ToonTalk itself. E.g., a teacher was conducting activities with birds and nests, having been told that she should focus, computer-wise, on different combinations of birds and nests: 1 bird to 1 nest, N birds to 1 nest, 1 bird to N nests, and N birds to N nests. This she did, but only by discussing with children expected behaviour for these situations and trying them together with the children, to see what would happen. Theme at the kindergarten, that week: human races. Simply by providing, as a suggestion, that different houses could be customized to represent different countries/continents, and that birds could be used to send letters to those countries, and that this setting could be used for a “1-to-n”, “n-to-1” activity made her feel much more comfortable and able to provide further suggestions on her own.

During the weekly meetings, we found that teachers would welcome some quick reference cards on computer-based educational topics, in order to build their activities around those topics. I.e., one for birds, with the elements aforementioned; another for programming, with questions 1 to 5a; another for working with sensors, mentioning the need to focus on preparing different consequences to the same actions; and so on.

While we still don't have results from this setting (it is still occurring), we can state that most results will be inconclusive regarding approach comparison, because approaches 1 and 2 (vd. section 2.4) proved more demanding than expected, in terms of teacher preparation and training, and in terms of time required for activity design and set up.

7. Framing of Future Research

We believe that the experience acquired along the past years, briefly described in the previous sections, will allow us to initiate more substantial research, after more solid theoretical and bibliographic grounds are established. However, such research will likely have a narrower scope than the questions and areas that we currently envisage. For this reason, we would like to summarize the ones most clear to us.

- Computer Science-wise, there is a need to identify specific concepts, applicable at the 3-year to 5-year olds age level (in order for children to explore such concepts, these should be translated into ToonTalk-specific activities).
- Autonomy-wise, there is a need to determine how much can 3-5 year old children achieve (perhaps as a group) without much adult intervention, given diverse approaches to computer-science activities, i.e., find approaches that better allow children to set off on their own;
- Complexity-wise, much research is needed: currently, children used as much as two robots (passing a ball between two copies of the same robot, have one robot build houses with a different robot inside them). How can this be expanded? And bird communication? Can children predict/expect results from robots in other houses? (All these questions are but examples of adding complexity.);
- Method: as mentioned in section 2.4, can programming activities be started right away? Or should children simply start by exploring the environment without programming, customizing it as they would a Lego world? And what about results from cause-consequence activities, for the child to try and grasp (or predict/expect) events, instead of controlling all that is happening? And are the answers different for different children?
- Integration within kindergartens: how much time can or has to be devoted to ToonTalk, in order to achieve visible results? How can other computer activities be included, without increasing computer time overall? Can guidelines for computer-programming kindergarten education be achieved?

8. Final thoughts

While much research has been performed on computer programming with children, only a small part of this was devoted to kindergarten education, possibly as little as 1% (Klein et al., 2000). And if we consider ages 3 and 4, not just 5-year olds, the amount of research is bound to be even scarcer.

Most research focuses on programming as a "black-box" activity, not distinguishing programming activities among themselves. However, computer programming is a varied and growing area, with many different features and accordingly different educational and learning aspects. In order to allow computer and programming activities to blend with the educational context and other educational themes, these features and aspects should be addressed, rather than ignored.

9. Acknowledgements

We would like to thank the cooperation of the team of computer-activities teachers, and also to the regular teachers in all kindergartens. Above all, we would like to thank the children involved, by their enthusiasm with ToonTalk and by being ever more demanding. Many of our most crucial insights were obvious and intuitive to you.

10. References

Degelman, Douglas; Brokaw, Ellen J. and Free, John U. (1986), *Effects of Logo Experience and Grade on Concept Learning and Creativity*, Journal of Educational Computing Research, 1986, Baywood Publishing Co., Amityville, NY, USA, Vol. 2, pp. 199-205. <http://www.vanguard.edu/faculty/ddegelman/logo.pdf> [March 27, 2003].

Haugland, Susan W. (2000), *Computers and Young Children*, ERIC Clearinghouse on Elementary and Early Childhood Education, University of Illinois, Champaign, IL, USA. <http://ericece.org/pubs/digests/2000/haugland00.pdf> [March 26, 2003].

Ingalls, Dan; Kaehler, Ted; Maloney, John; Wallace, Scott and Kay, Alan (1997), *Back to the Future: The Story of Squeak, A Practical Smalltalk Written in Itself*, Proceedings of the 1997 ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages & Applications (OOPSLA'97), Atlanta, GA, USA, October 1997. <ftp://st.cs.uiuc.edu/Smalltalk/Squeak/docs/OOPSLA.Squeak.html> [March 27, 2003].

Kahn, Ken (1996), *ToonTalk™ – An animated programming environment for children*. Journal of Visual Languages and Computing. (Abbreviated version in Proceedings of the National Educational Computing Conference, Baltimore, MD, USA, 7 (June): 197-217, 1995.) ftp://ftp-csli.stanford.edu/pub/Preprints/tt_jvlc.ps.gz [March 27, 2003].

Klein, Pnina S.; Nir-Gal, Ofra and Darom, Efraim (2000), *The Use of Computers in Kindergarten, With or Without Adult Mediation; Effects on Children's Cognitive Performance and Behavior*, Computers in Human Behavior, Amsterdam, Netherlands, Vol. 16, Issue 6, November 1st 2000, pp. 591-608. <http://web.macam98.ac.il/~nirgal/a-publish&study/artical-klein&nir-gal&darom.htm> [March 27, 2003].

Logo Foundation, (2000), *What is Logo?*, <http://el.media.mit.edu/logo-foundation/logo/>, Logo Foundation, New York, NY, USA [March 27, 2003].

Ministério da Educação, Departamento da Educação Básica (1997), *Orientações Curriculares para a Educação Pré-Escolar*. (English translation included in: Ministério da Educação, Departamento da Educação Básica, 1998, *Early Childhood Education in Portugal*, Departamento da Educação Básica, Lisbon.) Department of Basic Education, Ministry of Education, Lisbon.

Ministério da Educação, Departamento da Educação Básica (1999), *Early Childhood Education and Care Policy in Portugal*, OECD Background Report, [http://www.oecd.org/pdf/M00020000/, files M00020266.pdf, M00020270.pdf, M00020272.pdf, M00020275.pdf, . M00020278.pdf and M00020280.pdf](http://www.oecd.org/pdf/M00020000/,files/M00020266.pdf,M00020270.pdf,M00020272.pdf,M00020275.pdf,.M00020278.pdfandM00020280.pdf), Department of Basic Education, Ministry of Education, Lisbon [March 27, 2003].

Morgado, Leonel; Cruz, Maria and Kahn, Ken (2001), *Working in ToonTalk with 4- and 5-year olds*, presentation at the "Playground - novos ambientes de aprendizagem" Seminar, Porto, Portugal, organized by CnotInfor. An edited version was accepted for publication at the e-Society 2003, IADIS International Conference, 3-6 June 2003, Lisbon, Portugal. <http://www.utad.pt/~leonelm/TTon4-5.htm> [March 27, 2003].

Morgado, Leonel; Cruz, Maria and Kahn, Ken (2002), *ToonTalk in Kindergartens – Field Notes*, ICTE 2002 International Conference on Information and Communication Technologies in Education, 4th volume in CD-ROM (awaiting publication), Badajoz, Spain. <http://www.utad.pt/~leonelm/papers/TTICTE2002/TTknf.html> [March 27, 2003].

Myers, Brad; Hollan, Jim and Cruz, Isabel, eds. (1996), *Strategic Directions in Human Computer Interaction*. ACM Computer Surveys 28 (December 1996).

Papert, Seymour (1998), *Technology in Schools: To support the system or render it obsolete*, Milken Exchange on Education Technology [Online]. Available on-line at: http://www.mff.org/edtech/article.taf?_function=detail&Content_uid1=106 [March 23, 2003].

Saraswat, Vijay (1993), *Concurrent Constraint Programming*, MIT Press, Cambridge, MA, USA.

Smith, David Canfield and Cypher, Allen (1999). *Making Programming Easier for Children*, in Druin, Allison, ed. *The Design of Children's Technology*, Morgan Kaufmann Publishers, San Francisco, California, USA.