# Tangicons: Algorithmic Reasoning in a Collaborative Game for Children in Kindergarten and First Class

Florian Scharf, Thomas Winkler, Michael Herczeg

Institute for Multimedia and Interactive Systems - University of Luebeck 23538 Lübeck, Germany

winkler@imis.uni-luebeck.de

## ABSTRACT

In this paper, we describe the use of Tangicons, nonelectronic physical programming cubes for kindergarten and first grade children. Tangicons have been developed with the help of kindergarten children during various sessions of observing, playing and talking to them. Most tangible computing environments are too complex for young children. We developed an appropriate educational environment on a pedagogical basis resulting in easy to use tangible bricks, integrated in a physical game. Tangicons are haptic programmable bricks for programming a sequence of operations. Their symbol design is related to real world objects. With Tangicons, children are able to learn first steps of programming in a playful way.

#### Keywords

Tangicons, tangible media, programmable environments, learning, iterative design, user centered design, evaluation

#### INTRODUCTION

Since more than seven years through the KiMM-Initiative [23] we are concentrating on the design and development, testing and evaluating of programming environments for school children. We investigated especially, whether an additional pedagogical benefit for elementary school children in the age of 7 onwards emerge and whether these young children learn to program interactive systems themselves [20] [21]. We ask the question, whether it is appropriate to teach kindergarten and first grade children, in the age of 5 and 6 to program interactive systems. And if it makes sense, how should the interface look like? In order to answer these questions, we analyzed a variety of developments of many research groups and build systems ourselves over the last 7 years.

It turned out, that in the majority of cases computing environments for programming are too complex for 5 or 6 year old children. To be more appropriate, programming has to be linked to the physical environment, to props [11] and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDC '08, June 11-13, 2008 Chicago, IL, USA Copyright 2008 ACM 978-1-59593-994-4... \$5.00

eventually to phycons, objects for tangible programming [6]. However, certain problems arise from this approach. Should specific interactive objects be created? And even then, are children in kindergarten or first class able to handle the highly abstract task? Dealing with these questions and examining solutions of other researchers we attempt to develop an appropriate educational environment with easy to use tangible bricks, integrated into a physical game. This paper describes the tangible user interface (TUI) called Tangicons, which has been integrated into a role-playing game, allows for fine as well as gross motor skills. The environment is designed for educational purposes, to introduce children of the age between 4 and 6 into programming, collaborative work and reasoning. It can also be extended for further educational purposes by adding more functions.

Children begin their lives without any specific abstract knowledge and therefore have to learn everything from scratch. They interact with the world to develop their cognition [1] [14] [16]. From the second year onwards, sometimes even earlier, they learn to speak a language, but only at about the fourth year they can handle the double representation and figurative thinking [2]. Tangicons make it possible to understand first steps of programming and therefore are the ideal teaching aids to show young children correlations between programmable bricks and their results. They can be used in different ways and provide an increasing level of difficulty. The first part of childhood is known as the oral phase. After that, children begin to feel and handle their surrounding. This behavior is very important to them. They have to negotiate and act by themselves in order to learn. During this stage they acquire language by connecting all kinds of objects to words. They begin to interact with the world and learn new languages by grasping things and naming them. At the end of the first stage of development, the preoperational phase, a child is able to reason. At exactly this stage, when children are still in the stage of understanding their surroundings by grasping things but also develop first thoughts about logic relations, the foundations for programming should be set. It is very important to foster this as early as possible, because everything a child really understands can later be applied to similar problems. As for the ability to speak a language without any accent, it is too late to set the basis for a new language

at an advanced age. The ability to understand the world by handling physical objects is decreasing continuously with late childhood or early adulthood. At a young age children use their sensor-motor skills to explore the world. They need the reference to the physical world in order to learn. In terms of the pedagogic of Célestin Freinet children understand (e.g. mathematics) by handling, arranging structuring and counting objects. Thus they have the ability to visualize mathematics and physically take hold of it. Physical objects, like stones and pebbles, which children can touch and explore with their hands, help them to refine and stabilize the already acquired knowledge. Renowned pedagogues like Friedrich Froebel. Maria Montessori, Célestin Freinet and Jean Piaget have propagated this approach to physical bound teaching for a long time. This knowledge about learning habits of children can be used to foster logic and abstract ways of thinking and even first programming experience. Programming is similar to learning an abstract language. If we start very early to teach first programming skills in a playful manner, it is easier for children to learn complex programming languages and reasoning later on. More and more elementary and nursery schools begin to teach languages at an early age and for the same reason a basis for computer scientific way of thinking should be set. Tangicons can support this process. We thought about a way to get kindergarten and first class children interested in programming. For the 5 to 6 year old children the Tangicons are kept very simple but upgradeable. In later versions the level of difficulty can be increased for older children step by step. Thus the younger ones can learn the basics and the older ones can learn more advanced programming methods.

This paper describes a first prototype of non-electronic tangible programming bricks, called Tangicons, which can trigger many functions with the help of a LEGO RCX and a digital camera. We begin with a description about related work in the fields of Ubiquitous, Visual, and Tangible Computing. Before we show the process and the results of the development of the Tangicons with teachers and kindergarten children, we show the structure of the whole system.

## **RELATED WORK**

## **Ubiquitous Computing**

Very important for programming with children is the fact, that we have to get rid of the computer in the sense of a machine developed for office work with a keyboard, screen and mouse. Computers are difficult to use and master for young children and standard applications have too many integrated functions, which will confuse the young users. 5 to 6 years old users feel overburdened by these computers, where still technology is in the visible centre of attention [13]. Tasks cannot be fulfilled in an efficient way, because the design of the devices has not been optimized for this purpose. During the late 80s Mark Weiser coined the notion of Ubiquitous Computing at the Xerox Palo Alto Research Center (PARC) changing the perspective [19].

Ubiquitous Computing expects that future technology will act unseen in the background. There is no need for the user to bother with the technical aspects. Instead of such complex general purpose computers there will be information appliances, each constructed for a specific purpose. These appliances shall be based on three important aspects. First of all they should be kept simple. The complexity of a tool should be in accordance with its purpose. That means that there is no need for further functions, which would only distract the users. Technology should disappear, because it is not important for the user of an appliance to see how it works, only that it fulfils its task. The second aspect is about versatility. All appliances should be designed in a way, that they grant the user new and innovative interactions and applications. The last but not least important aspect is pleasure. All devices should be fun using them [13]. Ubiquitous Computing takes the human being to the centre of interaction. The computer serves just as an extension of humans, designed for support [9]. What is essential for the adult everyday user is even more essential for children at 5 or 6 years. We need easy to use interfaces, but computers have difficult input devices like a mouse and keyboard. They prevent the children from doing collaborative work because only one or two children can effectively work together in front of one monitor. This is another reason that the computer has to disappear and act unseen in the background of a social setting.

## **Tangible Computing**

The highest level of programming is seen in complicated functional or object orientated programming languages like Java or C. They require a high level of knowledge and abstract thinking and are not easy to learn. There are also scripting and macro languages, which are easier to learn but are restricted and still quite complicated in their usage. Visual programming languages are said to be easier because of their iconic user interface. Each module or program part can be seen and directly implemented in the existing program. Because of the iconic character of each module, they are easily recognizable and their function seem to be clear at first sight. This kind of programming is much more suitable for young children but still quite complex. Today there are new possibilities given by miniaturized devices and wireless connections. They make the user independent of the main computer and open up to new options of input devices especially tangible ones. The rather new approach of tangible computing has been proposed mainly by Hiroshi Ishii [7]. The knowledge of visual computing is taken, simplified and modified to get a quite new experience of programming. You can see and touch your programming modules, which creates a totally different relation to a program. A tangible programming environment is usually not as complex in programming as a visual or even non-visual programming environment. Too many tangible bricks would make it too complex to handle and therefore unusable. Because of their simplicity, they are mainly used for educational purposes to inspire

children to do programming. The children define the sequence of statements and explore the results. Thus the computer does not function as a black box anymore [17]. There is a very good example for tangible interfaces by the game "On the design of Camelot, an outdoor game for children" [18]. Here, children work together in a collaborative way to build a castle with virtual resources. They run around a lot between the construction site and their building blocks and make use of modern technology that is hidden in the background. Children have a lot of fun playing this game. But here the black box is still present. The children cannot get to the bottom of the technique. Our approach is similar to the Camelot game but it focuses more on setting the foundation for algorithm and opening the black box for children.

There are many approaches concerning tangible computing and it is discussed whether it is useful or not, considering complex programming languages. It is very difficult or sometimes even impossible to take a general purpose programming language and make it tangible. But the design of a reduced tangible programming language is certainly sufficient for many educational purposes. Yet only a few studies refer to programming for children in the age of 5 or 6. The younger the children are, the more we have to leave abtractions and complexity. But even in its simplicity of structure, this kind of education provides a foundation for increasingly more abstract and complex tasks. Children have to program themselves and get involved with their own ideas [4].

## **Tangible Programming Devices**

Tim McNerney developed the "Tangible Programming Bricks" [10]. They are based on LEGO bricks with included electronics based on the Cricket technology. The Tangible Programming Bricks are an easy way to program. The user has the ability to program all kinds of everyday devices like microwaves or toy cars by sticking different programming cards into the bricks and piling them. These bricks include different parameter cards for the various actions. They are better suited for older children or adults than for kindergarten children because programming with those blocks is still quite abstract. A difference between this programming device and the next two presented devices to the Tangicons system results from the integrated electronics. Tangicons do not need expensive integrated electronic parts. They just need a base of electronic parts for image processing (a digital camera and imaging software) and they are arbitrarily extensible without much further expenses. A relatively new project of Michael Horn is Tern [6]. Tern is a tangible programming language for middle school and late elementary school students. It consists of wooden jigsaw puzzle like pieces that can be connected to each other. By choosing different shapes of connectors, the possibility of programming with syntax errors was reduced to a minimum because some illegal programming sequences cannot be made. Like Quetzal or Tangicons, Tern uses non-electronic tangible bricks that

have to be photographed and then are processed by a computer. The Tern language lets the students control virtual robots on a projection screen. FlowBlocks from Oren Zuckerman are digital "Montessori-inspired Manipulatives" that encourage children in the learning of abstract structures [22]. The computationally enhanced building blocks foster educational concepts related to counting, probability, looping and branching. Children can connect the blocks and see how the power flows from the generator block through their path. They can install lights and counters to visually enhance their program and see the results. There are three other projects that had influence on our Tangicons. One of them comes from Peta Wyeth who designed Electronic Blocks with integrated electronics for children between three and eight years. They are based on LEGO Duplo Primo Bricks and can be piled for triggering different actions. Like our Tangicons, every block has another outcome but there is not the possibility of laying a sequence of directives. Another very interesting project is the Display Cube project [8]. The Display Cube is a small wooden cube with integrated acceleration sensors for gesture recognition, gravitation sensors for recognition of the spatial state, and a display for each side, respectively. In field tests it was used for learning of vocabulary and to foster three-dimensional imaginative power. The cubes have no buttons for input and the only way of interaction is made through gestures like shaking the cube. Although not obvious at first sight, those gestures are very easy to learn. Cubes are known from early childhood and it is already clear that there may be information to be found on every side of it. Thus the users can handle them intuitively. Tangicons take advantage of this feature, too. The third project is AudioCubes. By moving around four cubes on a glass table the user of the interactive installation can shape his own soundscape. A camera is tracking the positions of the bottom icons of the AudioCubes. According to their position and icons the cubes can trigger different sounds like drums, base, lead or string in the corresponding positions of the room. The idea of cubes with symbols on them is similar to the Tangicons design, but instead of having a fixed camera below a table used for picture recognition, we use a small digital camera to keep the costs low and provide high portability.

## **Visual Programming**

It does not cost much time and knowledge to learn the currently available visual programming languages in the toy and game markets and even non-professional programmers can learn to visually program and have the unique chance to develop their own ideas within a short time. The game industry, for example has discovered the wide range of possibilities that open up to a wide range of users, who can program their own toys. It are not only children who use visual programming to modify their toys, even a lot of adults use the same tools as the children, which satisfy their needs. Very good examples of visual programming are LEGO Mindstorms Robolab and the new version of NXT software. They are sold worldwide and have many users from relatively young students to adults. Robolab is a visual programming language designed for programming the LEGO RCX. There are also some other visual programming languages like LogoBlocks or PicoBlocks that will not be discuss in detail here. They are used for the programming of microcontrollers, but they are too complex and abstract for young children. The symbols used are too abstract and there are too many of them. Visual programming has also several disadvantages because everything is programmed directly on a computer. Children with their small hands, having not yet developed fine motor skills, have difficulties in using mouse and keyboard for interaction. Also collaborative work is hard to realize when working on a computer. Therefore children are forced to work individually. Those disadvantages led to Michael Horn's project Quetzal at Tufts University. He observed students who programmed with the visual programming software Robolab and saw that they sometimes had difficulties with the input devices mouse and keyboard. He came up with the idea to make the interface tangible and developed a software based on some functions that Robolab implements and also tangible bricks that represent exactly the functions of that specific software. The bricks had no electronics inside, which made them quite inexpensive. They had small symbols on them for picture recognition, the so called Spotcodes. Spotcodes are round, black and white 42-bit codes with two bit-date-rings each capable of 21 bits [3]. They can be decoded by image processing software for further processing. The only expensive electronic parts needed were a computer and a digital camera. After connecting the bricks and forming a program, the students had to take a picture of it. The Quetzal software recognized the sequence of the Spotcodes in that particular picture, translated them into RCX code and transferred the code to the LEGO RCX. The Quetzal project gave us important ideas for our own project.



Figure 1: Quetzal bricks

#### Development

We thought about the purpose of our system, what tasks should be accomplished and what problems the users might encounter. The approach was different than for normal software projects. Usually there are users in need to solve problems or tasks, who ask for software that could help them. Here we have child users who have no specific goals. We want them to accomplish or learn with the help of our product. The Tangicons system is not meant to solve difficult problems. So far in this paper the importance of play has been mentioned at several points. It is not important for the system to serve a purpose but to motivate, to intrigue and to socialize children. We concluded that in our case the five points below are the most important ones for educating kindergarten children:

- 1. collaborative working;
- 2. programming;
- 3. reasoning and reflecting about acquired knowledge and skills;
- 4. training sensor-motor skills;
- 5. using multiple senses.

We took a closer look at the Quetzal project [5]. It was a good idea to make the programming bricks tangible, but programming did not seem to work for very young children. We thought about the reason and came to the conclusion that the icons he used were too abstract for young children. The tangible bricks represented exactly the visual version of the Quetzal computer program. Programming with Quetzal is very close to programming with Robolab. But young children have problems with deciphering those abstract objects because they do not have any relation to their real life objects. They need tangible objects, which can be recognized at once. We decided to adopt the idea of Tangible Quetzal with a new approach of making tangible programming bricks with a new pedagogic background.



Figure 2: Final Tangicon cube

In the center of our system are the Tangicons. They are influenced mostly by ideas of Froebel, Montessori and Freinet who mentioned the significance of practical work for children. Children learn by handling objects, grasping things and working actively on tasks with the help of all their senses. We wanted to create tangible bricks, which could be easily recognized even by very young children. They should appeal to different senses like visual, haptic, and auditory senses and have symbols close to real world objects. But more than just our planned tangible bricks were needed. By thoughts based on Montessori and Freinet, we came to the conclusion that we needed a given framework for the children within which they are allowed to follow their own ideas. They should not be restricted to a certain given exercise. We needed more than one kind of task, so that the children can decide themselves which kind of task field suits them best. It is best not to interfere too much in the learning process, but to provide the children with the needed materials. We give guidance with our tangible system, but do not take a significant role so the children still have the freedom to choose and to develop in their own way. We needed more diversity of tasks and also an appropriate attractiveness of the environment for the children was necessary. The learning effect is best if somebody is interested in what he or she is doing. In order to keep their attention on the bricks the children needed a product they could program and in which they were also interested. We wanted the children to have fun in their activities with the side effect of learning.

## GAME DESIGN

## "The Wizards Tower"

The materials used for the Tangicons system are inexpensive. For a full set of material for the Wizard's Tower game we need the following:

- 2 x 6 (12) Tangicons (including start and stop)
- 1 digital camera (wireless in future versions)
- 2 LEGO RCX (NXT in future versions)
- 2 x the same amount of building bricks for the towers (we used 2x 120 bricks)
- LEDs with different colors integrated into a box
- modified Quetzal software
- 1 laptop (or other computer) placed in the background







Figure 4: Box with 2 RCX and LEDs

The game is based on very simple rules. There are two groups of Wizards competing with each other. The two groups comprise of four children each. Every group has the same amount of bricks to build a tower and every brick must be used until there are no more left. The group, which finishes first building the tower, has won. There are some rules which prevent from making the building of the tower too easy. "The Wizard's Tower" deals with little wizards. The children slip into the role of modern wizards who get the instruction to build a tower for their mighty wizard parents. The wizard family who finishes the tower first will rule the wizard land. Thus the groups of competing little wizards try to enchant each other with the help of Tangicons. The enchantment of a group means that they are prevented from building. But they have the possibility to break the spell - again with the help of Tangicons.



Figure 5: Game setting

## The Game

For the game setting it is important to mind some aspects. Cognitive development is accompanied by motor development. The cognitive development process improves with exercising the game. Additionally, children have a great need for movement. Young children at the age of 4-5 cannot concentrate on cognitive work for a long time. They need to have breaks to move around. That is why there are stations of enchantment (where the Tangicons are programmed) and the bricks for the towers are on distant places. Each group consisting of 4 children has a base station for programming and another station as the construction site for the tower. Both programming stations are positioned side by side but none of the groups is able to watch the others programming. Nevertheless they can see how far the other group has set up the tower. Starting from the programming station, for every group there are the building bricks for the tower at a distance of about 5 meters. Between the base stations of both groups there is a box with the 2 RCX in it and 2 times 3 different colored LEDs. At the beginning, all of the 8 participants are located at their base stations. Each group has received a kit of 6 Tangicons, including start and stop. The Tangicons have different symbols. One group is selected to begin the game. This group is preparing together the enchantment with their Tangicons, while the other group has to wait for the blinking lights that are triggered by the spell. The wizards need their Tangicons kit for enchantment and disenchantment. By laying a sequence of Tangicons they can trigger an enchantment. For example they can lay the sequence: 1 second yellow light, sound, 5 seconds red light, 1 second blue light. The given number of Tangicons restricts the number of tasks to 4 at a time, because it is too difficult for the children to remember a more complicated sequence. Now the second group is enchanted by the spell and is not allowed to build the tower as long as the spell is not broken. To break the spell they have to remember the given sequence of the enchantment and try to code exactly the same sequence with their Tangicons. In the meanwhile the first group is allowed to build their tower. For building the tower, the children have to run about 5 meters to the station with the bricks. They are only allowed to carry one brick at a time to the construction site. If the spell of the second group is broken, this group has to enchant the first group in the same way they did. The groups take turns until one group finishes building their tower with all the given bricks and wins. Each part of the game was tested during the design phase with children. It was important to know in advance, how long the distance between the building bricks had to be, how many bricks each group needed and how many bricks the bringer was allowed to carry at a time for a good workflow. We observed the children building the towers to see how long it takes and if programming with the Tangicons in-between building was possible. It was important to see how the children get along with each working step and if there were still problems within the game model. With this method we could react to the behavior of the young users and quickly adjust to their needs within our design process. This formative approach saves a lot of work and is better than changing a system after completion. During the game the children learn to understand that there is a connection between the programming bricks and their self made program. Watching them play with a prototype for the first time, we were surprised that one kid understood the idea of programming with the Tangicons at once after a brief introduction. He took the start cube followed by a sequence of lights and one sound and a stop cube, and explained to us what will happen. Then he explained it again to the other three children around him.

## **EVALUATION**

The evaluation took place in the Lauerholz elementary school of Luebeck with the "Tiger" kindergarten children and also with a first class of the Paul-Klee primary school of Luebeck. We worked with interviews, because the children were of course not able to read or write, and with written notes that were taken during play. We also placed a set of 2 video cameras, which recorded the scene permanently for later evaluation. During evaluation we spoke to all children (5 girls and 3 boys at Lauerholz elementary school and 4 girls and 4 boys at the Paul-Klee elementary school) separately so they would not bias their answers on other kid's opinions.

## Lauerholz School

We discovered that our technically experienced kindergarten children made incredibly fast progress during play. In the beginning they behaved very restrained and were not sure what to do. First they needed about 4:30 minutes for casting a spell with our help. Then they could not concentrate on the blinking lights and sounds and tried to do something with the cubes, not knowing exactly how to handle them. While giving them little instructions, they managed to "break the spell" in about 3:10 minutes. Only after a short period of time after these first attempts they realized how to use the Tangicons and the time to decipher the spells and reconstruct them was shortened. The first try took about 50 seconds (not including picture taking of about 40 seconds) but was unfortunately wrong. The time to fix this mistake was only 12 seconds for programming and 20 seconds for taking the picture. Bewitching the other group still took about a minute (including taking the picture). The children thought about their spell very properly instead of saving time. During the second round the children became faster. The time for breaking the spell, including picture taking, dropped down to between one 1:00 and 1:20 minutes and was solved mostly at first try, though it is important to mention that these values depend on the kind of Tangicons used for creating this spell. If only "10second-light" spells are used, it is impossible to accomplish such times. The time for casting a spell to the others dropped down to about 30-40 seconds (including picture taking) because the children where able to find even a tradeoff between time consumption and difficulty of their spell. These results show, that kindergarten children are very fast learners who can understand the connection between our Tangicons and the LEDs. At first they had some difficulties in remembering the sequence, and had to look up after every step but they improved very fast, although they sometimes still had to look them up for a second time. 7 out of our 8 children said that the Wizard's Tower game was a lot of fun and they liked it very much. After the game ended, they even asked if they could play it again. Only one boy said that it was ok, but he did not want to play it again. He claimed that the running part was too hard for him. From the position of an external observer it sometimes looked like not all of the children had much fun. At the beginning there were only two children programming in the first group, while the other two members of their group watched them. After a while one of these children started watching the code again and telling the others about the sequence when they forgot it. The fourth child seemed to have fun in running and building. They all found their parts within the group. The second group became collaborative in another way. All four members counseled about programming steps. Two laid down the Tangicons and the others gave advice. All of them worked together at each part of the game. The level of difficulty was experienced as medium hard and not too difficult. One girl even said that the game was easy for her. When we asked what kinds of steps were difficult, all of them agreed, that running was the hardest part. Two girls and one boy said that they had difficulties handling the digital camera but did not regard it as a real problem. The truth is, we helped them out to photograph. Strangely enough nobody claimed to have had problems with conjuring even not after inquiring.

## **Paul-Klee School**

At the primary school, we had a whole class consisting of 28 people of which the teacher picked out 8 children for our game randomly. The other 20 children were sitting on benches as spectators to the left and right side of the game set. Like at the other school, the children also needed a little time to understand the rules of the game. But the groups were very different. While the first group learned very quickly to decipher the spells by programming and reduced their time constantly, the second group had more problems. The first group needed 3:24 minutes for their first try, reduced to 2:05 minutes with their second try, 1:17 minutes with the third try and then they solved the problem in 0:29 minutes. Not only the reduction of the time needed was amazing, also the number of attempts for breaking a spell. They had to correct their spell during the first attempt three times, the next round two times and after that, they

solved everything immediately. This shows, that they really understood what they had to do and what they must pay attention to within a short time. The second group had a good start with 1:35 minutes to decipher the spell, but after that it took longer. The second attempt was 3:17 minutes and they had to correct the spell two times and the third round seemed to be very confusing to them. They needed 5:47 minutes and five corrections. After that guite long attempt, they seemed to be more confident in what they were doing in their next round and they also reduced the programming time to 3:14 minutes and in the last round to 2:00 minutes. During their last two rounds, they also seemed to have more fun playing the game. It took this group a little bit longer to get the idea behind the programming part but after that, they improved very fast. Despite the fact that children have different intellectual perception and therefore the groups are not equal, the scenario shows that both groups improved a lot during play. One thing we noticed was that the spectators got to know the game play quite fast only by observing the others. They sometimes said to the players what they had to do and gave them advice. That gave us the idea, that it is even possible for larger groups to play the game without much preparation. The groups can be switched easily, because the spectators already know how to play the game, only by observing the others and therefore do not need further introductions.



Figure 6: Playing the game at Paul-Klee School

Like the children of the kindergarten, all of them wanted to play the Tangicons game again. We even had to play 20 minutes longer during their break and their next class. Two of the boys, and three girls said the Tangicons game was great. The other three children thought it was good. Except one girl and one boy who just enjoyed playing a little, all of them had much fun playing the game. Even the boy who found the game itself moderate had fun playing it. The difficulty of the game was experiences as moderate by one boy and two girls. For the rest of the children it was quite easy to handle. When asking them, what they found difficult about the different game parts, the given answers also where similar to the other school. Three girls and one boy said that running was very strenuous and one boy said that also the programming was not very easy for him. Besides some technical problems, the game was very well accepted

by the children, which manifests in the fact that they even wanted to spend their break in order to play a bit longer. With a more fluidly game play the children would probably be even more enthusiastic about it.

## CONCLUSIONS

As a result it seemed that the children themselves were pleased with the game, except for a few technical difficulties. One of the most disturbing errors resulted from handling the digital camera very abrupt. We did not know about this problem until the children played with our Tangicons. When adults handle a camera they take a little more time with it. In our case the 1/125 sec. shutter time was absolutely sufficient and worked for us adults in several tests without a problem. Children on the other side put the camera down immediately after having shot their picture. In that case, the 1/125 sec. shutter time results in very bad picture recognition with approximately 80% failure. Therefore the shutter time had to be decreased to at least 1/200 sec. or more. But then the taken picture gets darker and it is also hard to recognize all of the Topcodes. A higher picture resolution would probably help a little, but doubles the processing time to about 15 seconds. However, the second disturbing part we discovered was the communication between our hardware. We did not expect that the USB cable from the digital camera and even more the infrared communication between the LEGO RCX and the tower could be disturbing. The cable was sometimes hanging in front of the camera and we had to help the children with that. As children are very agile, they often tend to jump between the tower and the RCX, disturbing the communication. After telling them not to do so, we could go on, but it mutilated the natural flow of the game a little. The internal speakers of the RCX are too quiet. The children had difficulties in telling when the code was uploaded to the RCX and also often missed the beeping sound of their spell. It would be better not to use the internal speakers or replace them by others. At last we would also recommend not to use such a heavy and big digital camera like the one we used (Canon Powershot S40). Because of their small hands, it is possible for children to handle this camera, but with some difficulties.

Besides some need for improvements in game play and wireless connection, the Tangicons project was accepted by its young users. The evaluation shows, how fast the children learned by improving their gaming abilities with Tangicons. The game we invented is just an example for what can be done with Tangicons. One can totally change the story and actions according to certain goals. It is only restricted by fantasy, inventing other games with Tangicons that trigger effects, move objects or react to input. Tangicons can be used for many different purposes related to programming for young children. For example the whole Tangicons gaming structure could be changed for using the Tangicons blocks for memory tasks, where long sequences must be memorized. Arithmetic functions are implemented in the Quetzal program too and just have to be adapted to the Tangicons system, if needed. All in Quetzal predefined and later added functions can be used with Tangicons but we decided to stick to programming for kindergarten children with our prototype and therefore keep it simple. The graspable user interface of the Tangicons gives personal multi-modal experiences by handling the wooden cubes and therefore supports a deeper understanding of programmable technology through actions, which are abstract (virtual) and physical (real) at the same time. In this setting the computer (the standard box with keyboard and screen) disappears completely [15]. It is invisible to the children while programming. The children themselves dominate the computer: they write a simple program in lining up the physical objects as defined by physical icons (phycons). Because of the individual learning with digital, interactive media in an playful way, involving gross- and fine-motor skilled activities and in context symbolic and rudimental algorithmic thinking and acting there is not only a high motivation for the children and a better understanding of abstract information, but also a natural reference to individual experiences and actions in collaboration with the other children.

#### ACKNOWLEDGMENTS

We would like to thank Michael Horn for providing the Quetzal software. It saved us a lot of work.

## REFERENCES

- 1. Brosterman, N. (1997). Inventing Kindergarten. *New York: Harry N. Adams.*
- DeLoache, J.S., Uttal, D.H., Pierroutsakos, S.L. (1998). The Development of early Symbolization: Educational implications. *Learning and Instructions. Vol.8, pp 325.*
- 3. de Ipina, D.L., Mendonca, P.R.S, Hopper, A. (2002). Trip: A low-cost vision-based location system for ubiquitous computing. *Personal & Ubiquitous Computing*.
- Druin, A., Montemayor, J., Farber, A., Simms, S., Churaman W., D'Amour, A. (2002). Physical Programming: Designing Tools for Children to Create Physical Interactive Environments. *CHI 2002*.
- 5. Horn, M. (2006). Tangible Programming with Quetzal: Opportunities for Education. *Master's thesis, pp 66. Boston: Tufts University.*
- 6. Horn, M.S., Jacob, R.J.K. (2007). Tangible Programming in the Classroom with Tern. *Proceedings of CHI* 2007 ACM Human Factors in Computing Systems (CHI Trends Interactivity), ACM Press.
- Ishii, H. (2007). Official Homepage of Hiroshi Ishii of the Tangible Media Group. *Boston: MIT. Available at:* http://web.media.mit.edu/~ishii/
- 8. Kranz, M., Holleis, P., Bilandzic, M., Vetter, J., Schmidt, A. (2006). The Display Cube as Playful TUI

To Support Learning (Video). *The 4th International Conference on Pervasive Computing*, 2006.

- 9. McLuhan, M. (1964). Understanding Media: The Extensions of Man. New York, 1st Ed. McGraw.Hill, reissued 2003. Corte Madeira: Gingko Press
- 10. McNerney, T.S. (2000). Tangible programming bricks: An approach to making programming accessible to everyone. *Master thesis. Boston: MIT Media Lab.*
- 11. Montemayor, J. (2003). Physical Programming: Tools for Kindergarten Children to Author Physical Interactive Environments. *Dissertation, University of Maryland.*
- 12. Montemayor, J. Druin, A., Faber, A., Simms, s. Churaman, W. and D'Amour, A. (2002). Physical Programming: Designing Tools for Children to Create Physical Interactive Environments. ACM Conference on Human Factors in Computing Systems, CHI 2002.
- 13. Norman, D.A. (1999). The Invisible Computer: Why Good Products Can Fail, the Personal Computer is So Complex and Information Appliances Are the Solution. *Boston: MIT Press.*
- 14. Papert, S. (1990). Children, Computers and powerful Ideas. *New York: Basic Books*.
- 15. Paradiso, J. (2005). The disappearing Computer. Communications of the ACM, Vol. 48, No. 3 (March 2005), pp 70. New York: ACM Press.
- 16. Piaget, J. (1975). Das Erwachen der Intelligenz beim Kinde. *Stuttgart: Klett Verlag.*
- Resnik, M., Berg, R., Eisenberg, M., Turkle, S., and Martin, F. (2000). Beyond Black Boxes: Bringing transparency and aesthetics back to scientific investigation. *Int. Journal of the Learning Sciences, Vol. 9, No. 1.*
- Verhaegh, J., Soute, I., Kessels, A., and Markopoulos, P. (2006). On the design of Camelot, an outdoor game for children. *IDC 2006. New York: ACM Press.*
- 19. Weiser, M. (1991). The Computer for the Twenty-First Century. *New York: Scientific American*.
- 20. Winkler, T., Kritzenberger, H., Herczeg, M. (2002). Mixed Reality Environments as Collaborative and Constructive Learning Spaces for Elementary School Children. *Denver: ED-MEDIA 2002.*
- 21. Winkler, T., Herczeg, M., Höpel, I., Reimann, D. (2004). Learning in our increasing digital world by connecting it to bodily experience, dealing with identity and systemic thinking. *Atlanta: SITE 2004.*
- 22. Zuckerman, O., Arida, S., Resnick M. (2005). Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives. *Portland: CHI 2005*.
- 23. KIMM: http://www.kimm.imis.uni-luebeck.de