Tangible storytelling: let children play with the bits

Pedro Rito[1], Ana Loureiro[2]
[1] ESE - Polytechnics Institute of Viseu, Portugal, rito@esev.ipv.pt
[2] ESE - Polytechnics Institute of Santarém / CIDTFF - University of Aveiro, Portugal, acclloureiro@gmail.com

Abstract: The use of tangible objects makes it possible to create interactions, or dynamics, which are alternatives to the mouse and keyboard in the process of communicating with the computer. The construction of these objects incorporating electronic components lets us bring that momentum to another level. This meeting with the technology allows children to take an active role, while there is a purpose of control over the objects, which becomes important to them. With the reinforcement of that control, the introduction of programmable digital electronic components also allows the child to develop, strengthen and feel the impact of their role as competent designer and creator of technology. Current technology allows the construction of these objects and the communication with computers at a low cost through micro-controllers, using, on one hand, the open source software and on the other the open hardware.

Introduction

Children express stories they listen by different ways. With the help of an object or a set of objects they can build their own storytelling, as long as they feel in control of those objects (Montemayor, Druin, Chipman, Farber, & Guha, 2004). With the use of familiar physical objects children reinforce their knowledge, through exploratory and acquaintance activities (Scarlatos, Mbogho, & Jaworska, 2006).
Available technology allows that technical operations - considered as medium to high demanding such as text, video or sound manipulation - can be made by children and also permit that communication computer interfaces go beyond the traditional interaction through mouse and keyboard. At same time computers can be replaced by other devices with similar characteristics (Montemayor *et al*., 2002).

One of the examples of the above mentioned technology is the Arduino based microcontrollers, which are open source tools, can be programmed and therefore established interaction with computers. The use of these types of microcontrollers allows the environments and processes of creativity enhancement, since the user doesn't need to waste time with details related with technical function but it can be used for experiment and discover new environments and scenarios. The programming is made using friendly software similar to IDE Arduino (IDE - Integrated Development Environment). Although there is the need of coding to teach the microcontroller, this specific software allows the communication with other software, like Scratch. Being an easy learning hardware, arises the opportunity to build tangible interfaces where a child can easily work (Wyeth & Purchase, 2002).

Open Source Hardware

Open Source Hardware (OSHW) is used to define tangible artifacts – machines, devices or other physical gadgets. OSHW's design is publicly available in order that anyone can make, modify, distribute and use it («OSHW - Definition of Free Cultural Works», nd). Due to this flexibility and openness various solutions can arise, although based on the original design.

An object described as OSHW allows people to have the freedom to control the developed technology and at same time to share the knowledge they gather in an open manner («OSHW - Definition of Free Cultural Works», nd). People are encouraged to share the knowledge about the software development and its applicability.

Arduino is an example of OSHW. It was developed by Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino and David Mellis and publicly presented in 2005. This specific microcontroller permits the establishment of communication between tangible and digital objects. The communication is settled between two environments and it is designated by physical computing («The Open Source Definition (Annotated) | Open

Source Initiative», nd). Environments that trigger this type of communication are related to two worlds - digital and physical world. The transmission of information can be observed at a lower level, resulting in a conversion of physical energy to electrical energy which is perceived by computers. This energy conversion is called transduction (Noble, 2009).

Arduino

Arduino is a microcontroller that allows on one hand to create and developed other types of interfaces to communicate with the computer (beyond the traditional mouse and keyboard), and on the other hand the creation of objects - physical or not – able to react to the information emitted by the computer.
Arduino's main characteristics (Duemilanove 's version) are: fourteen input/output digital ports, six analogical ports, communication is made through a serial protocol and the fact it only needs a single power supply - can be a USB (Universal Serial BUS) port or a nine volts battery. Six of the fourteen digital ports are PWM (Pulse With Modulation), e.g., ports that allow the signal to vary within a certain range, while the remaining ports only receive values as zero and one.

Multiple versions of the Arduino were developed by the same team, whose main differences and features are physical size (occupied space), the number of available analog and digital ports and the way to communicate. One of those examples is the Arduino Lilypad, developed by Leach Buechley and the Canadian company SparkFun Electronics, and presented to the community in 2007. This specific model allows that a microcontroller can be integrated and interact with e-textiles or that it works as a decorative object (wearable). Lilypad has about eighteen centimeters, can be stitched to a e-textile piece and linked to other sensors or similar parts (Igoe & O'Sullivan, 2004). This way anyone can create their own e-textile pieces and creativity gets expanded.

Arduino's Operation

Arduino's hardware has open source software associated, with the same name of Arduino IDE - this software is compatible with every hardware version and derived from the IDE for the Processing programming language. Arduino IDE is written in Java language and based in the Wiring project. In this IDE coding lines are written and encapsulated, in order that anyone can understand and work with them, even not being an expert on those languages. Processing is a programming language developed by Ben Fry and Casey Reas based on John Maeda's work entitled Design by Numbers. It is an easy to understand language and with that visual artist or newcomers unfamiliar with programming can develop their work without limitations or restrictions.

The microcontroller, apart from receiving instructions from a computer, can also receive information from sensors (input signal) and send information to a computer via actuators (output signal) (Banzi, 2008). Both sensors and actuators are electronic devices that allow Arduino to communicate with physical world. As example of sensors we can mention some tangible objects such as buttons, movement sensors, temperature sensors, CMOS (complementary metal–oxide–semiconductor) sensors, infrareds; and as actuators we can mention the led or the engines. Arduino not only receive information from sensors but also from other software. Examples of that software are the Processing, Adobe Flash, Max/MSP or Scratch. Those mentioned software can not only send data to Arduino but also have the ability to receive it. Arduino, once taught, doesn't need to be connected to a computer, it can act alone, e.g., can be created an interface that communicates with the computer via the Arduino and this can be powered from a battery.

Arduino's Version

There are also other versions of Arduino, some are compatible with the Arduino IDE or with other software. Some examples of such micro-controllers: the Paperduino developed by *Guilherme Martins*, the Freeduino the Boarduino, the Sanguino or the Funnel I/O. As connections to these type of micro-controllers there are Shields, which are cards that allows to expand the Arduino, for an expansion of space-related work (for more connections) or with the inclusion of hardware that was not originally designed on the micro-controllers.

Although it was initially designed for Artists, Architects and Designers, we can find different situations where new features for this type of micro-controller are being created and used, such as its application in robotics (Balogh, nd), in music (Vallis, Hochenbaum, & Kapur, nd), prototyping the interfaces for games (Bekker, Hummels, Nemeth, & Mendels, 2010) or programmable material for children (M. Eisenberg, Elumeze, MacFerrin, & L. Buechley, 2009).

Software Programming

According to Eisenbeis *et al* there has been some emphasis on the discussion towards children's programming, particularly on the construction of programming languages suitable for children, on children’s need to learn some particular topics related to programming or even on the main theme, related with the need or not of children learning to program (M. Eisenberg, Elumeze, MacFerrin, & L. Buechley, 2009). However, these authors emphasize that learning programming for children can be a reinforcement to their creativity in mathematics, science and art.

ToonTalk’s Software

For Kahn (Kahn, 1996) learning to use computers without learning to program is like learning to read without learning to write. This author and his team developed ToonTalk, a program that allows children to program using animated objects, where the work environment is a computer game. With the emphasis on creating a programming environment enjoyable and fun for children, ToonTalk allows games to be created and at the same time it helps control the sensors and motors of Lego, with no need on teaching people about programming

Scratch’s Software

There are some experiences on the use of open software and open hardware with children, as in the case of the Dialando (Smith, 2010) which was developed through the use of Processing, Arduino and Scratch, an interface that allows children learning to program using this free technology and physical objects that serve as low cost physical interfaces. Scratch was developed at MIT Media Lab and it is a learning software for programming through an interface of individual blocks. The use of a programming syntax that makes use of blocks makes the learning more explicit than implicit (Richards & N. Smith, 2010). For these authors, the use of Scratch allows students to be able to get the basics to start work in any programming language.

Linked to this software there is the Picoboard, which is a micro-controller that has several embedded sensors on the same plate, enabling communication with the Scratch software. Based on the Picoboard, the SenseBoard was developed (Richards & N. Smith, 2010), with the concern of adding new features, namely the inclusion of LEDs, the possibility of working with more complex programs, the possibility of writing over the Internet and connecting motors. The project GogoBoard, on other hand, presents a micro-controller with similar features but includes a kit using electronic components that can be purchased at any store. Both Picoboard, SenseBoard or GogoBoard have similar capabilities to the Arduino, but the main potential of Arduino is that is not exclusive to a single software and it is not directed only to a certain type of users.

Construction of Tangible Interfaces and Storytelling

In the eighties Mitchel Resnick and Stephen Ocko developed a project called LEGO / Logo that allowed the integration of Legos pieces with programming, through the language Logo. With motors and sensors robots or other objects can be built, which were taught and could function without being permanently connected to a computer. These authors view allowed the physical computing was deepened while the use of a programming language with Legos fostered children’s creativity (Millner, 2009).

Currently the use of electronic components, sensors and actuators is not expensive, while the physical space they occupy allows these components to be hidden, decorated inside and inside of objects whose physical dimensions are suitable for children (M. Eisenberg, Elumeze, MacFerrin, & L. Buechley, 2009). We have no interest in furthering our work to the level of ubiquitous computing (interacting with the computer only with invisible actions), because we want children to build their interfaces or the objects of their narratives and also want to ensure that children feel the absence control.

To Montemayor *et al* the use of fragile technology by children, as is the case of some sensors and actuators, does not allow them to build consistent cognitive behaviors and predictable behaviors. We must also consider that when technology requires a certain degree of precision to work or is interrupted by a malfunction, that can distract the child by withdrawing the control. Therefore, the objects used should be reliable, durable, robust and flexible technology-based (Montemayor, Druin, Chipman, Farber, & Guha, 2004).

Today we find some examples of the use of electronics with children: Blackbox (Andersen, 2008), experience with children from preschool, where they are invited to interact with a box with sensors and LEDs; Storyroom (Montemayor, Druin, Chipman, Farber, & Guha, 2004), where children, aged between 4 and 6 years, interact with various objects, including Lego Duplo blocks, which incorporate sensors and are used to construct and tell their stories; the Electronic Blocks (Wyeth & Wyeth, 2010), for children between 3 and 8 years, allowing them to make use of Lego Duplo blocks that contain electronic circuits and with the line-up of these blocks they are able to program and build robots and other objects, animals, among others.

Further Work

Using the Arduino has major advantages, by being an open hardware, being a micro-controller compatible with multiple operating systems, including Linux, having a low market price and an interface that enables communication with other software. The learning curve is quick and we expect to get almost immediate results with their use. Being free software, the libraries needed to be adopted are also available and are free. The use of programmable objects, e.g., targeted programming environments for children and the constant involvement of the teachers in learning will, as stated Eisenberg *et al* (and with which we agree) leads children to construct an enriched narrative, art and simulations and provide a daily presence, inside and outside the classroom, in their intellectual.

It is not our interest to discuss the question of the advantages or disadvantages of being free or not, but to foster in to the community the knowledge that exists and can be developed and used in several contexts. We have tried throughout this document to highlight the main features and valences of tangible objects, so that the community has an interest on using and expanding this network of knowledge.

In the near future we would like to continue to explore the issues presented here and provide training with a group of Masters students of basic education, future teachers, so they can develop this project during the probationary period, creating activities relating to physical computing and building digital storytelling tangible with children between 4 and 8 years old.

Acknowledgments. 1 Co-funded by PROTEC and PTOFAD programmes since April 2010. 2 Co-funded by the FCT and the European Social Fund (ESF) as part of the Programa Operacional Potencial Humano Europeu (POPH) of QREN, since 01 January 2009 to august 2010 and co-funded by the PROTEC programme since September 2010

References

Andersen, K. (2008). Black box: exploring simple electronic interaction. Proceedings of the 2nd international conference on Tangible and embedded interaction (p 207–208).

Balogh, R. (no date). Educational Robotic Platform based on Arduino.

Banzi, M. (2008). Getting Started with Arduino (1st ed.). Make.

Bekker, T., Hummels, C., Nemeth, S., & Mendels, P. (2010). Redefining toys, games and entertainment products by teaching about playful interactions. International Journal of Arts and Technology, 3(1), 17–35.

Buechley, Leah, & Eisenberg, Michael. (2008). The LilyPad Arduino: Toward Wearable Engineering for Everyone. IEEE Pervasive Computing, 7(2), 12-15.

Eisenberg, M., Elumeze, N., MacFerrin, M., & Buechley, L. (2009). Children's programming, reconsidered: settings, stuff, and surfaces. Proceedings of the 8th International Conference on Interaction Design and Children (p 1–8).

Igoe, T., & O'Sullivan, D. (2004). Physical Computing: Sensing and Controlling the Physical World with Computers (1st ed.). Course Technology PTR.

Kahn, K. (1996). ToonTalkTM--An Animated Programming Environment for Children. Journal of Visual Languages & Computing, 7(2), 197-217. doi:doi: DOI: 10.1006/jvlc.1996.0011

Millner, A. (2009). Computer as Chalk. Massachusetts Institute of Technology.

Montemayor, J., Druin, A., Chipman, G., Farber, A., & Guha, M. L. (2004). Tools for children to create physical interactive storyrooms. Computers in Entertainment (CIE), 2(1), 12.

Montemayor, J., Druin, A., Farber, A., Simms, S., Churaman, W., & D'Amour, A. (2002). Physical programming: designing tools for children to create physical interactive environments. Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves (p 299–306).

Noble, J. (2009). Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks (1st ed.). O'Reilly Media.

OSHW - Definition of Free Cultural Works. (no date).. Retrieved 22 February 2011, from http://freedomdefined.org/OSHW

Richards, M., & Smith, N. (2010). Teaching UbiComp with sense. Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, NordiCHI '10 (p 765–768). New York, NY, USA: ACM. doi:10.1145/1868914.1869024

Scarlatos, L. L., Mbogho, A., & Jaworska, M. (2006). Teaching with Tangibles: A Tool for Defining Dichotomous Sorting Activities.

Smith, A. C. (2010). Dialando: tangible programming for the novice with scratch, processing and arduino.

The Open Source Definition (Annotated) | Open Source Initiative. (no date). Retrieved 23 February 2011 from http://www.opensource.org/osd.html

Vallis, O., Hochenbaum, J., & Kapur, A. (no date). A Shift Towards Iterative and Open-Source Design for Musical Interfaces.

Wyeth, P., & Wyeth, G. F. (2010). Electronic blocks: Tangible programming elements for preschoolers. IFIP TC. 13 International Conference on Human-Computer Interaction (Vol. 1, p 496–503).

Wyeth, Peta, & Purchase, H. C. (2002). Tangible programming elements for young children. CHI '02 extended abstracts on Human factors in computing systems, CHI EA '02 (p 774–775). New York, NY, USA: ACM. doi:10.1145/506443.506591