Scratch + CNC: Mixing Physical and Digital Worlds

Dirceu Maraschin Jr[1], **Anderson Gutierres**[1], **Tiago Thompsen Primo**[1]

[1]CDTec – Universidade Federal de Pelotas (UFPel)
Postal Code – 96.010-610 – Pelotas – RS – Brazil

{dmaraschin,aesgutierres,tiago.primo}@inf.ufpel.edu.br

Abstract. *Knowledge retention occurs most effectively when there is experimentation and there is an association between understanding what was learned conceptually and a real experience. We propose the development of an extension for Scratch 3.0 allowing the interaction with a CNC (Computer Numeric Control) machine. A proposal of cooperation between two environments in the learning context is created, combining programming in blocks with equipment able to provide outputs in physical objects.*

1. Introduction

The globalized world inserts the technology in everyday life directly or indirectly, but it does not occur in the same way within the classroom in most Brazilian schools and when it happens the greatest potential is not extracted. Thus, despite the latest advances in the offer of technology to transform teaching and learning in the traditional classroom environment [del Rosario 2012, Kirkwood and Price 2014], there is a general concern about the effectiveness of current approaches, which fail to generate clear evidence that technology produces results that contribute to the educational environment. Therefore, perceives that there are failures in the educational model where the technological evolution was not inserted in the school context, occurring a gradual reduction in the engagement of the learners faced by the educational methodologies which become obsolete.

It is needed encouragement for change to happen even in front of the difficulties and, in this process, we believe that the first step, even small, reflects more actions that generate positive results and best practices. In this thought, the present work has as an objective to propose the development of an extension to the programming environment in blocks Scratch version 3.0 where there may be interaction with a CNC equipment from which they can be concretized in different materials and in different ways what has been programmed in the tool, such as figures, geometric shapes, traces and others.

We intend to achieve different competencies listed in *"Base Nacional Comum Curricular (BNCC)"* [Brasil 2017] and others aspects of the creative learning methodology [Papert 1980, Zhang and Yang 2013] connecting concepts already approached in the classroom as physics and mathematics, for example, to develop logical reasoning through programming (computational thinking [Wing 2006]) related to the process of assembly and running of the machine (electronics, robotics, etc.), adding the "maker spirit". This set of experiences serves as the basis for the development of a teaching and learning methodology in which many skills can emerge.

2. Learning Environment

The use of technologies to support the teaching-learning process is not new. It stimulates teachers and students through technological components that help in the materialization of

DOI: 10.5753/wave.2020.212084

abstract concepts in phenomena that help in the continuous improvement of this process. New challenges are constantly arising with the demand for new approaches reflecting social changes and the evolution of technology for more complex systems.

[Wing 2006, Resnick 2012] defines computational thinking as the capacity that an individual has to initiate a process of elaboration and solution of problems. The logic of programming allows the coherent articulation of the information in a logical sequence that makes possible the resolution of problems, being one of the fundamental knowledge for the construction of computational thinking [Ramos and da Silva Teixeira 2015]. However, these teachings are transmitted in isolation most of the time, without contextualization and relation with other areas of knowledge and problems of the real world. This practice contradicts David Ausubel's Significant Learning Theory, with a concept of Derivative Subscription [Ausubel 2012], for example. Thus, computational thinking can assist in the development of adjacent skills such as (i) abstract thinking, with different levels of subjectivity in understanding the problem; (ii) logical thinking, in the formulation and elimination of hypotheses; (iii) algorithmic thinking, constructing solutions to problems in different steps to arrive at the most efficient and effective solution; (iv) scalable thinking, dividing a problem into smaller parts to compose a complex solution [Andrade et al. 2013].

In this sense, research has been carried out to introduce concepts of computation in the context of Brazilian primary and secondary education. Parallel to this comes the promotion of Creative Learning initiated by [Papert 1980] through the concept of constructionism, having its in-depth research by [Resnick 2014]. Based on the adoption of the four P's (Projects, Peers, Passion and Play), aiming at stimulating students by placing them at the center of the educational process, enabling them to plan, create and test in real situations of daily life, acting in an active way before social problems and the themes that involve them [Gomes dos Santos and Galembeck 2017]. Aligned this idea, we wish to put in practice the development of this proposal by extending the computational thinking approach to creative computing.

3. The Propose

Motivated by the previous discussion, the principles of the present work focus on the proposal to develop an extension to the Scratch tool that allows the communication between the software and the CNC machine interface, allowing the user to transform what was programmed into a real object. The wanted development scheme of this extension is illustrated by Figure 1.

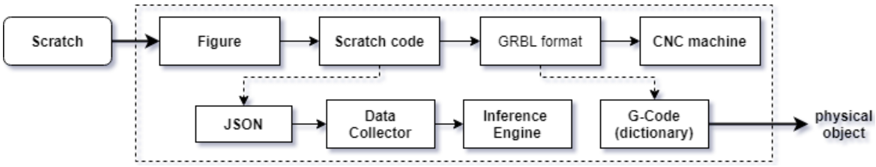

Figure 1. Scratch-CNC extension scheme.

In this schematic, we can observe the Scratch tool as the main module of the system and the physical object as the final product of the process. From this, the process must follow some well-defined steps:

i. Scratch data collector: all the block programming performed in the Scratch workspace interacts with the *scratch-blocks* and *scratch-vm* modules. Through since last, all code is constructed and executed from the blocks, being normalized to a *JSON* structure, where each event can be accessed and interpreted independently. Thus, a structure is needed for the correct collection and processing of necessary information:
   - data collector: this module should be responsible for collecting properly the relevant information present in *JSON* structure built by the Scratch tool in backend;
   - inference engine: should select and apply the most appropriate rule in each step as the application is executed, constituting the previous step to the translation to *G-Code*.

ii. Code translation: from the access to the *JSON* structure, the code must be translated into another format that can be interpreted by the CNC interface, (*G-Code*).

iii. Communication between interfaces: after the collect and processing of the code it's necessary send to CNC machine the code that will represent the drawing.

vi. Object creation: the final step of the process will be the materialization of the drawing made in Scratch in a perspective of physical-virtual interaction.

It is emphasized that in the first moment we wish to work only with simple geometric forms (such as straight lines, circumferences, and polygons) that can be expanded later. It's possible to imagine several teaching-learning possibilities associated with concepts of mathematics and physics with formulas and measures, for example. Besides that a universe of concepts regarding the operation of the CNC machine integrating concepts of mechanics, electronics, programming, and others beyond the possibility of building the machine itself with reused materials. A brief description of the two main components of the project, the Scratch program, and the CNC machine is given.

3.1. Tools

The Scratch tool is a visual programming language developed by a partnership between MIT and University of California. This is not the first environment and programming language intended for beginner programmers. Scratch replaces the code typed with a "drag and drop" approach inspired by LogoBlocks[1] and eToys[2]. The tool emphasizes media manipulation by supporting programming activities that are of interest to young people, such as creating animated stories, games, and interactive presentations. Programs are built by assembling stacks of colored command blocks that eliminate syntax problems and encourage the exploration of the environment.

In the 1940s MIT created the NC (Numerical Control) which later evolved into CNC with the advent of computing. The computerized numerical control by itself comprises a system that allows control of machines through a specific code called *G-Code*, through which it can interact simultaneously with electronic and mechanical components, such as the axles moved by motors and the operating head. Used mainly in the large-scale industry, this technology has become part of a range of smaller applications that can be built indoors reusing materials and diversifying the applications as machines that can draw, paint and write combining mechanical parts for drive and electronics for control.

[1] www.media.mit.edu/projects/logo-blocks/overview/

[2] www.etoys.com

4. Considerations

We propose in this article the implementation of an extension for communication between the Scratch tool and a CNC machine. The objective is that the student has the experience of transforming what was programmed into a physical output. Besides that, many teaching-learning skills can be associated such as (a) understanding of coordinates, (b) concepts of electronics, (c) programming logic, (d) multiple dimensions and many others. For the versatility of the machine we can list possibilities such as (i) creation of drawings from mathematical functions; (ii) printing of characters by cutting from the student's drawing to compose a board game; (iii) markings on wood, paper and other materials; (iv) creation of electronic circuits through the production of PCBs with the CNC operating as grinding. We understand that the inclusion of technology associated with experimentation can be sources of greater engagement and retention of knowledge in the classroom environment.

References

Andrade, D., Carvalho, T., Silveira, J., Cavalheiro, S., Foss, L., Fleischmann, A. M., Aguiar, M., and Reiser, R. (2013). Proposta de atividades para o desenvolvimento do pensamento computacional no ensino fundamental. In *Anais do Workshop de Informática na Escola*, page 169.

Ausubel, D. P. (2012). *The acquisition and retention of knowledge: A cognitive view*. Springer Science & Business Media.

Brasil (2017). Base nacional comum curricular (bncc). educação é a base. *MEC/CONSED/UNDIME*, page 600.

del Rosario, M. (2012). Ict in education policies and national development. In *Post-Secondary Education and Technology*, pages 17–38. Springer.

Gomes dos Santos, V. and Galembeck, E. (2017). Por uma ciência para o dia a dia: possibilidades para aprendizagem criativa e significativa na educação básica. *Enseñanza de las ciencias*, (Extra):4035–4040.

Kirkwood, A. and Price, L. (2014). Technology-enhanced learning and teaching in higher education: what is 'enhanced'and how do we know? a critical literature review. *Learning, media and technology*, 39(1):6–36.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.

Ramos, F. and da Silva Teixeira, L. (2015). Significaçao da aprendizagem através do pensamento computacional no ensino médio: uma experiência com scratch. In *Anais do Workshop de Informática na Escola*, volume 21, page 217.

Resnick, M. (2012). Reviving papert's dream. *Educational Technology*, 52(4):42–46.

Resnick, M. (2014). Give p's a chance: Projects, peers, passion, play. In *Constructionism and creativity: Proceedings of the Third International Constructionism Conference. Austrian Computer Society, Vienna*, pages 13–20.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.

Zhang, L. and Yang, H. (2013). Definition, research scope and challenges of creative computing. In *2013 19th International Conference on Automation and Computing*, pages 1–6. IEEE.