# A playful tool to introduce lower secondary school pupils to recursive thinking

Violetta Lonati, Dario Malchiodi, Mattia Monga,
Anna Morpurgo, and Manuel Previtali

Università degli Studi di Milano, Italy, http://aladdin.di.unimi.it

**Abstract.** To introduce lower secondary school pupils to recursive strategies, we developed a software tool for supporting active guided explorations of the execution of a recursive algorithm.

**Keywords:** recursion, K12 CS education, tool for algorithm exploration

## 1 Introduction

Recursion is a fascinating topic that can provide a powerful approach to problem solving, but is often considered to be out of reach for most educational contexts. Thus, we aimed at supporting a proper demystification of recursion in lower secondary schools, in order to suggest it as one of the many tools even younger people can bring on the table of problem solving. We designed a piece of software around a sort of *little people metaphor* [3] in which recursion is presented as a delegation of self-similar sub-tasks to "helpers", that we called "fairies".

## 2 A tool to visualize recursion

The tool, called `Fatine` (the Italian word for 'little fairies'), visualizes the computational steps of a recursive algorithm that computes the reverse of a string chosen by the user, and is designed to support a work of analysis and abstraction about how the algorithm works. `Fatine` displays a "computer" whose monitor shows some moving objects, and the role of the latter is to represent the underlying process in some concrete way: – a sequence of circles represents the recursion stack, each circle corresponding to a call of the recursive function (as many circles as characters in the input string); – the substring passed as argument of a call is represented by a tower under the circle corresponding to the call and its height is proportional to the substring's length; – in the first phase of the process, when calls are made, the tower shrinks while moving from left to right, and when subsequently values are returned and composed into the solution, the tower moves back to left and it grows; – the function is carried on by a little fairy: she detaches a piece from the top of the tower, passes on the tower, receives the tower, attaches a piece at the bottom of the tower, waits, sleeps. `Fatine` is organized in "levels" as common in video games. The pupils can observe the execution of the algorithm from view points of increasing depth. In the first level the computer is a "black box" simply receiving the string as input and returning its reversed form. In the next levels pupils can conduct experiments by pausing the process execution: to promote abstraction and generalization, they are not allowed to pause and observe the behaviour of consecutive function calls. When an experiment ends, they can try again with different strings.

## 3 Experiment and evaluation

We tested the tool within a learning unit devoted to $8^{th}$ graders and having the following goals: (1) know the fundamental features of recursion (2) be able to execute a simple recursive procedure described in natural language (e.g. raise 2 to a given power); (3) be able to identify the fundamental features of a recursive algorithm, given its description or the possibility to observe its execution; (4) understand that a problem can be solved by solving subproblems linked to one another. We designed our learning unit according to our *algomotricity* [1,2] approach, thus we provided a proper environment to foster mental models of the topic under investigation through "unplugged", motoric activities and the use of tangible objects and then engaged pupils in a "plugged" (software supported) activity to foster abstraction and conceptualization. In the "unplugged" activity pupils executed a recursive algorithm to compute the length of a string represented by a LEGO tower, each pupil executing a function call by following the instructions on a note. The plugged activity was based on `Fatine`: working in pairs on a computer, pupils saw how `Fatine` runs the string-reversing algorithm, and they could pause the execution in order to observe characteristic aspects of the process, more and more specific as they proceeded. The tool supported them in understanding what is going on during the execution of the recursive algorithm, by getting them aware of what happens at each single step of the process and meanwhile grasping, at a higher level, how such steps are interrelated. Then the two algorithms (string length and reverse) were compared by reviewing their key features and highlighting their commonalities. During the discussion, all the basic features of structural recursion emerged. In particular, the pupils inferred and told the teacher that the instructions in the two activities, the ones on the notes handed out and the ones executed by the fairies but not visible, had to be similar. The pupils had also to work out what these instructions were and write them down, a task which required them to rework on what they had learned.

## References


1. Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A., Torelli, M.: What you see is what you have in mind: constructing mental models for formatted text processing. In: ISSEP 2013. pp. 139–147. No. 6 in Commentarii informaticae didacticae (2013)
2. Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A., Torelli, M., Zecca, L.: Extracurricular activities for improving the perception of informatics in secondary schools. In: ISSEP 2014. LNCS, vol. 8730, pp. 161–172. Springer (2014)
3. Harvey, B.: Computer Science Logo Style. MIT Press, second edn. (1997)