

Programming is a complex intellectual activity. We observed through the years that is difficult for some students to understand the logic of a program and to familiarize themselves with the control structures. In order to help smooth this initial relationship with programming, we tried Scratch. We analyze the use of Scratch in two Computer Science 1 (CS1) courses at ORT Uruguay University: one in university degree and other in vocational studies. We use Scratch in the very first weeks with the purpose of improving students' programming experiences and motivation, and also to detect its influence, if any, in scores and dropout rates in comparison with normal courses.

Particularly, we applied Scratch for 3 weeks of our 15 weeks courses. In the first week we presented the basic concepts of programming (variables, loops, conditions). We developed a guide to introduce the main elements of Scratch and exercises. The exercises -clearly defined and proposed in order of incremental complexity- were oriented to formalize and to understand the ideas of process and control. We did not include threads and events. We used only one sprite.

The second week we proposed more "intellectual" exercises: the cat (one of the icons provided by Scratch) "thinks" numbers and wants to know its sum, product, mean and maximum. We introduced variables. In the lectures, the exercises were solved by hand for each student and, after that, were discussed and coded in Scratch in the computer's class. In the lab session, each student programmed himself or herself the solutions and solved the additional exercises proposed.

In the third week, we included more exercises to reinforce the main ideas. We introduced Java (university)/Visual Basic (vocational). We made a parallelism between some constructions in Scratch and the language.

We applied tests and contrasted the results with normal courses and found that students who used Scratch expressed higher motivation but there was no statistical evidence of differences neither in dropout rates nor obtained scores. A possible interpretation is that the students of the Scratch course have two "jumps": first they must familiarize with Scratch and then with Java/Visual Basic.

We found some opportunities to improve Scratch in order to simplify its use in CS1. The lack of boolean variables can be supplied by integer variables acting as a flag. The input of user data is a limitation, but we can use random generator and suppose that the program "thinks" a number instead of asking for one to the user.

With the aim to establish a correspondence between the concept of "message passing" and object oriented programming, we suggest that the broadcasting scheme would be directed to a specific item, not only general.

The Spanish translation is confusing in some topics. For instance, if we have a variable "suma", the expression "change suma by 30", translated as "cambiar suma por 30", is not clear. If suma equals 10, that expression changed suma to 40 but, in Spanish, one is induced to think that the final value is 30, not 40.