

Preconceptions of novice learners about program execution

Sylvia da Rosa

Instituto de Computación

e-mail: darosa@fing.edu.uy

Universidad de la República

Montevideo - Uruguay

Introduction

Our research seeks to connect high school and university teachers. The aim is to help students understand how a program is executed by a computer. The activities of this study were carried out with four groups of high school students, aged 13 to 15, in Uruguay. The point of departure is that a program is both a formal text (an algorithm) and a physical object executed by a computer. Hence, two different didactic approaches are required:

- for algorithms and data structures written in programming languages a didactics similar to that of mathematics
- for the program as an object executed by a physical device a didactics centred around the actions that take place when the program is executed.
- What do students think about the correspondence between the algorithm and its execution?

The problem

The assignment statement clearly reflects the correspondence between the algorithm and its execution:

- from the point of view of the program text the assignment gives a value to a variable (a mathematical variable)
- from the point of view of the program as executable object the assignment alters a memory cell (a physical object)

... our subjects seem to have guessed,..., that the questions were about change rather than mathematical equality... (Dehnadi, S., Bornat, R., The camel has two humps)

The study

Hence, the study is based on a problem involving assignment. Our methodological framework is Jean Piaget's epistemological theory.

- First the students solve a problem and write their solution as an algorithm.

What about the execution of their algorithm?

- Second, we borrow some methods from computer science unplugged to make the students play the roles of the different hardware components when running the program.
- Third, students anticipate the result of running a program, then run the program and compare the results with their predictions.

Writing the program

We work with drinking glasses and liquids of different colours. The problem is to exchange the contents of two of them without mixing them.

- The sequence of glasses simulates an array, and the problem simulates switching the values of two cells.
- The students are divided into groups of four students and each group works with a sequence of glasses. The groups work independently.
- The purpose is that the students make the exchange using an auxiliary glass.

The glasses are divided into sequences of eight glasses each, enumerated from 1 to 8. The liquids can only be poured into glasses made of glass as 1 and 7. Some of the glasses are full. The task is to exchange the green and red liquids without mixing them.



Writing the program

All the groups succeeded and then questions were posed to help the students to describe the following algorithm:

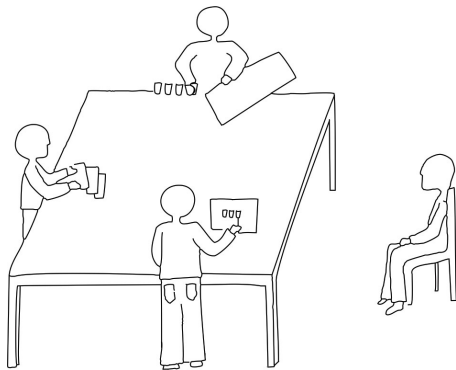
1. show state of glasses
2. pour contents of glass[5] into glass[x]
3. pour contents of glass[2] into glass[5]
4. pour contents of glass[x] into glass[2]
5. show state of glasses

In the next activity this is the program to be executed by students playing the roles of a simplified computer.

Runnig the program unplugged

- Three students act as the different components of a simplified computer (CPU, ALU/Memory and Display). The fourth student will be the user.
- This activity is inspired by "Computer Science without a computer" (csunplugged).
- The purpose is to give the students a small taste of how these parts interact when running a program and to highlight the fact that computers simply follow instructions, without understanding the whole process.

The glasses are hidden behind the piece of cardboard. By executing a show type instruction, ALU/Memory turns the cardboard so CPU can see the state of the glasses. By executing the "poor contents" instructions ALU/Memory selects an adequate glass and makes the exchange (hidden).



The computer runs the program

Once the cs-unplugged activity has been successfully completed, the students worked individually in the following activities:

- to anticipate the results of small Python exchange-programs (the students have been instructed in basic Python).
- to run them and compare those results with the predictions they made.

In order to compare students that have experienced the unplugged activity and students that have not, this part was also done with two groups that have not participate in the previous activities.

Our focus was on the answers of the students about the changes of values of the variables. However, we found some remarkable results in students answers about the behaviour of the programs.

Case 1	Case 2	Case 3
print v	v[0] = v[2]	print v
v[0] = v[2]	v[2] = v[5]	v[0] = v[2]
v[2] = v[5]	v[5] = v[0]	v[2] = v[5]
v[5] = v[0]	print v	v[5] = v[0]
		print v

Here, we describe one of the questions regarding the behaviour of three programs:

Given the assignment $v = ['w', 'w', 'g', 'w', 'w', 'r', 'w', 'w']$ ¹ and the programs above, do you think that in each of the cases the values of $v[2]$ and $v[5]$ are exchanged when executed?

¹ $v[i]$ is the value of the list in position i , $0 \leq i \leq 7$

Ans/std	All the parts (26 st.)	Third part (31 st.)	Total (57)
No	18 (69%)	14 (45%)	32
Yes	8 (37,7%)	17 (54,8%)	25

- 57 students answered the question: Does the exchange take place?
 - of the 26 students that participated in cs-unplugged: 18 said "no" (69%) and 8 said "yes" (37,7%)
 - of the 31 students that did not participate in cs-unplugged: 14 said "no" (45%) and 17 said "yes" (54,8%)
- "No" responses correspond to cases 1 (print sentence at the beginning) or 2 (print sentence at the end). "Yes" responses correspond to case 3 (print sentence at the beginning and at the end).

Some conclusions

- The analysis of the results is preliminary and several questions have arisen. However, it is to some extent clear that the students that had participated in cs-unplugged, attribute to the print sentence the power of making the exchange.
- Thinking that a sentence has an impact beyond the sentence itself has been mentioned by other authors, but here the point appears related to a *human way of thinking*.
- Their thinking is coherent with the fact that in the cs-unplugged activities, the students tended to show the state of the glasses *after each of the sentences of the exchange* as if only by showing it takes effect.

Synthesis and further work

Towards a didactics of programming.

- A didactics of algorithms and data structures: how do we help students in describing a general algorithm to solve a problem.
- A didactics of programs as physical objects: how do we help students in constructing knowledge about the correspondence between the actions they describe (an algorithm, formal text) and the actions that *a computer will do* (a program, physical device).
- This study is a first step. We have to find theoretical explanations about students' ideas revealed in the study.