COULD YOU HELP ME TO CHANGE THE VARIABLES? COMPARING INSTRUCTION TO ENCOURAGEMENT FOR TEACHING PROGRAMMING

Dimosthenis Makris, Kleomenis Euaggelopoulos, Konstantinos Chorianopoulos, Michail Giannakos

Michail Giannakos, Marie Curie/ERCIM Fellow at NTNU
Research goal

- To build an understanding of the main interactions between children and the tools in creative activities and to consider improvements on the current processes

Felicitous, creative and collaborative environments
Contemporary computer education has emphasized in the “consumption” (i.e., the use) of technology instead of the creation and understanding of it.

Video game development gives learners a “mental workout” and allow them to develop a number of cognitive skills.

Learner-centered education
Video Games for Learning: Beyond Playing

- This time it’s personal: from consumer to co-creator
- Beyond passive learners

They feel comfortable playing with interactive games, animations, and simulations, but not creating their own. It is as if they can “read” but not “write.” Resnick

Most schools around the world teach programming with passive way. Students learn the principles of computer programming similarly to History, without the freedom to create something on their own, without discovering that programming with accessible visual platforms like Scratch can be creative and amusing.
Mitra & Dangwal (2010) proposed to let students learn, on their own, with the help of a friendly mediator, who had no knowledge of the subject. They found that students’ performance was comparable to children who learned the subject from qualified teachers. Their findings suggest that any topic from simple internet browsing and language learning, up to molecular biology might also be facilitated by a friendly and encouraging mediator.
It is not something new

- Constructionism
- Do It Yourself (DIY) philosophy
- Maker communities
- Hackerspaces
- Modding
- Reggio Emilia
- Learning by Design
- End User Development
Challenges:

- Whether and how the Mitra encouraging pedagogic style is applicable to teaching programming in secondary education?
- How does the Mitra style of teaching affect students’ attitudes and self-confidence towards programming?

Opportunities:

- Students’ learn programming in a more meaningful way
- Change the idea of what computing is in K-12 education
Field Study

- Corfu: 24 students
- Second Grade of Gymnasium (Middle School)
- 13-14 years old
- Two different groups (Control-Experimental)
- Same teachers', same context and content
Groups

- Sugata Mitra (Modding) Group
- Traditionally Instructed-Control Group

Measures

- Self-Assessment Manikin (SAM) is a non-verbal pictorial assessment technique (adopted by Bradley & Lang, 1994; @ Journal BTEP)
- Students’ attitudes towards programming (adopted by Giannakos et al., 2013; @ SIGCSE)
The 1st Group: traditional Instruction

- The teacher used the Scratch tool to demonstrate to the children how to change variables in the game.
The 2nd Group: Sugata Mitra

- The second group learned about the same programming concepts in an encouraging style.
- The teacher did not provide any instruction about the variables theory, but encouraged the students to change the score and the ship-health features of the game in Scratch.
- The students indirectly worked with the same variables concept.
Results (1/2)

- Continue programming courses
  - Traditional: 35%
  - Sugata Mitra: 80%

- Continue using Scratch
  - Traditional: 55%
  - Sugata Mitra: 100%
Results (2/2)

Self-Assessment Manikin (SAM)
Before the experiment 90% of both groups were positive that “they know what programming is”

After the experiment, only 45% of the Traditional group were positive, as opposed to Sugata Mitra’s 100%

Teacher observations are in alignment with assessment results
Teaching approach seem to influence students’ attitude towards the programming course.

Students in the traditional instruction group are less likely to get involved with programming in the future than the participants in the encouragement group (probably because their experience was less stimulating).

Our findings (assessments and teachers’ observations) suggest that once the students had the opportunity to be creative, confident and unique, their interest in programming increases.

The encouragement approach with a visual tool results in greater intention for future involvement.
Limitations and Future work

- Small sample size
- Short duration experiment
- Limited to attitudinal measures, other measures like cognitive skills will allow us to better understand the benefits of encouragement teaching style
Thanks for listening

Questions?

Contact: michail.giannakos@idi.ntnu.no
WWW: http://www.idi.ntnu.no/~michailg/