

How we taught Children Robotics and Programming

Olivia Schütz
Department of Electrical Engineering
HTL Weiz
Weiz, Asutria
schuet200302@sr.htlweiz.at

Abstract—Robots and modern technology are now significant components of everyone's lives. But do young people understand technology, and how can we make difficult ideas understandable? The project's goal was to determine the best way for children between the ages of 10 and 12 to learn a subject related to robotics and later apply that knowledge directly. To do so, we divided each class into groups of 3 or 4, and we subsequently compared how quickly each group could program particular concepts. The findings indicated that the groups where the mentor first explained everything in detail and on a level, they could understand, then left them with small goals that would eventually form into bigger and bigger programs, were the ones that would be the quickest finished and motivated to start new, more complex programs, like a computer game in some ways.

Keywords—robotics, children, effective learning, basic concepts, little goals

I. INTRODUCTION (HEADING 1)

The future of electronics will be in the hands of children. They regularly use technology, whether it's on their computer, iPad, or smartphone. But the intriguing aspect is that, with a few notable exceptions, fundamental ideas about robots or what they are, are not well understood. Therefore, the likelihood that kids will be inspired to learn about it and later study this subject and reinvent a concept or make it better increases if we can bring robots, coding, and technology closer to children in a much more profound way of understanding them. However, to start, our own experience has shown that the best outcomes come from treating people with regard, speaking to them directly, and breaking down this potentially difficult subject into manageable pieces, much like a video game. To go much further, we will first enumerate the current applications of robotics and programming languages, the later one we used and the upgrades we made, how we explained the subjects, how we dealt with reluctant students, and finally, the conclusion.

II. CONCEPT & DESIGN

A. Introducing Robots

The primary goal is to provide them with something relatable. The original plan was to provide them with just the robot's parts and instructions so they could gain first-hand expertise. Lego bricks would also be included, because many kids have already constructed and programmed codes using Lego bricks [1]. They would learn concepts like analog or digital sensors with the robot they would be using, which is important, because we will later create programs that require both.

B. Difference between Servo & Motor

The objective is to use metaphors to teach them basic ideas like servos and motors, referring to illustrations that describe

how the topic works. We agreed that it was a good idea because it is simple to comprehend. But in addition to helping them visualize the idea in their minds, we would also start a program on the robots. Depending on what you attach, you would press either the button which reads servo or motor in this program. Later, you would see a circle with a red line. The kids could move to any spot in the circle along this red line and watch the wheels turn.

C. Programming Language C

For the programming language, we would use C, because of its block similarity. Through metaphors and not going too deep into the topic, we thought it would make a terrific learning experience. The metaphors were again little pictures, but this time the kids would also have a little sheet of paper on which the main concepts, like function and loops would be written down. It would be an asset when they are coding by themselves. To let them know the structure of the program we would guide them through the starter program "Hello World". Explaining what the main is, how it could be processed, what the return is, and once again not going into depth because there just would be confusing.

D. Motivating Children

Not everyone can get motivated, but by being friendly, talking eye to eye, and making the learning process engaging, a majority would be occupied at least a bit. Because of the approachable atmosphere, children would be most likely to ask the mentors questions and do not stay in their silence. To make the learning process interesting we taught to assemble it like a video game. A big task is broken down into smaller tasks in which the child gets appreciation and praise, when it completes their job.

E. Dealing with Problems

If there was a problem with building up the machine, we would directly help them, because most of the time they would not know which screw to use or where to put them. If there was a problem with the code, we would show them the line in which the mistake was made and gently direct them in the right direction. But not telling them the answer. If there would be a dispute, we would try to calm the situation down, talk to them, ask why this happened, and give each child a different task to do, so they can cool off and stop thinking about the argument.

III. IMPLEMENTATION

A. Explanation

Like previously mentioned, we did adopt the concept of using metaphors. In the field of programming, we explain loops and functions. And in the field of components, we explain the features of a motor as an example. Besides that, we held a short presentation in which one of our team would

ask the class questions like, which robots do you have at home, when were the first robot invented, and which characteristic of a human being is to compare with one of the robots?

B. Task

The two main tasks which we gave them were constructing the robot with instructions and writing a program with the help of a flow chart. The flow chart would show what the program should do. Each mentor would then either show them the flow chart and explain to the children what they should do. And later on, going through the concepts again, and maybe writing the first program with the group and if there is a need helping them. Or showing them the flow chart, giving them the sheet of paper, where the functions are listed with the explanation, and throwing them into cold water and looking if they make something out of it.

C. Helping

As we mentioned in the section concepts, we did follow the ideas of being nice, making small talk, and praising them if there were refinements. And when there were arguments, we tried to separate them and deviate the children with small duties, like asking them if they could bring you a screw.

IV. CONCLUSION

We cannot stress this enough. The young are our technology's future. They will comprehend new or similar concepts they already learned easier if we teach them how computers and machines work at a young age. In addition, some people may develop an interest and a desire to pursue a subject related to robotics. Those youngsters may rethink ideas and revolutionize contemporary technology. But how should we teach children? We should, first, make a comfortable atmosphere in which they can *thrive* and ask a question. Secondly, we must explain to them the basic ideas in a way they understand them. Meaning using metaphors and examples from day-to-day life. In the first practical experiences, the children should have someone again explaining the concept and showing them it on an example. When this phase is over, they should try out programming themselves. If they make an improvement, it is the mentor's responsibility to praise them and nurture their curiosity.

REFERENCES

- [1] Rollins, M. (2014). Beginning Lego Mindstorms Ev3. Apress.