

Robots Byte In: An Exploration of Computer Science Education in Middle Schools

Mansi Gupta, Marwa Nur Muhammad, Shikha Prashad

**Faculty Advisor: Douglas S Blank
Bryn Mawr College, Bryn Mawr, PA**

Goals & Purpose

Recent studies have shown that computer science as a field of study for undergraduates has declined significantly. Not only has it been difficult to attract students to this field, but it has been equally difficult to retain them. One of the primary reasons behind this may be the fact that students are not introduced to computer science at an early stage. It would be ideal to introduce computer science along with the natural sciences to provide the students the possibility to explore the subject further. It is also essential that the first impression is stimulating and engaging.

Previous research has mostly focused on undergraduate introductory courses, whereas our project targets middle schools. The Institute of Personal Robotics in Education (IPRE) has designed a curriculum for the introductory course and developed software called Myro which uses robots as a learning tool. Myro is written in Python and aims to make programming easier for beginners. We extended IPRE's curriculum to develop and implement effective learning strategies specifically aimed at middle school students. We investigated whether the early introduction of computing and the use of engaging activities involving robots impacted their learning curves positively.

Methodology

Our course was eight weeks long through which we illustrated the basic elements of programming using the Scribbler robot. This robot was chosen because of its low-cost, robustness and compatibility with Myro. It also includes IR sensors and has the ability to detect obstacles. In addition, it has a camera and allows a marker to be attached to the robot to draw. We preceded our course with an initial survey in which we explored their knowledge and perceptions of computing prior to taking our course. We asked similar questions in our post survey to compare how the course affected their understanding.

Each week, we met for two hours in the computer science lab at Bryn Mawr College with thirteen students between the ages of seven to thirteen. This age group was younger than expected; thus the course was adapted accordingly. Each of the classes incorporated a final activity that allowed the students to apply what they have learned that day. We concluded every session with a survey which provided the students with an opportunity to give us feedback about that class. To aid the learning process, we created a text which contains essential syntactical information to remind the students of necessary details. The text acted as a constant guide for them and is designed to make the course self-explanatory. The text is divided into parts to accompany each session. The chapters were organized in the following way:

Hello Scribbie: an introduction to the Scribbler robot and its control by a gamepad.

Activity: Steering the robot around a solar system setting.

Left, Right, Scribble: teaching the basic movement commands. *Activity:* Have the robot draw different shapes using a marker on the robot.

Mission Variation: an introduction to variables using Madlibs; introducing the commands to speak, take and save pictures; learning how to use a file to create a sequence of commands. *Activity:* drawing shapes using a file.

Finding Byteland: a review of all concepts covered so far. *Activity:* drive robot inside three unknown zones using movement commands; take pictures inside zone and work with the class to solve the mystery.

If, Else & Again: the ideas of conditionals and loops were introduced through an

interactive board game that we designed where the students were the board pieces and which required students to complete various educational challenges.

Loop (If Else & Again, 2): Python syntax for conditionals and loops was introduced. *Activity:* Programming the robot to steer in a room-sized maze spread out through the lab; programmed movements to draw shapes using loops.

Beep, Beep, Beep: introducing the concept of music, frequency and sound. *Activity:* using the concepts to create music using the beeping abilities of the Scribbler.

Final Review: students were given the freedom to use whatever they have learned and they worked in groups to give final presentations.

In order to make Myro more accessible, we added functions that masked the complexities of Python from the students. For example, to make the robot move in a square, the students will be taught to use a loop. Instead of writing the following,

```
def square():
    for i in range(4):
        forward(speed, time)
        turnLeft(speed, time)
```

the program can be condensed to writing:

```
loop(step, 4)
```

where step is defined as:

```
def step():
    forward(speed, time)
    turnLeft(speed, time)
```

Defining loop in this manner simplifies the programming and solidifies computing concepts. Since they will be using actual commands from the Python language, it will expose them to simple programming, but will not force them to understand the intense syntax involved.

As the students learned programming, they also gained insight into how it can be applied to other subjects including physics, music, English and geometry as our chapters describe.

Results & Conclusion

While our sample size was small, the results appear promising. However, we noticed that some of the students did not take the surveys seriously and a small sample size probably distorted the results to a larger extent. The pre and post survey comparisons show the following. At the end of the course:

- More students strongly agreed that **computing is fun**.
- More students strongly agreed that **girls can do computing**.
- More students agreed that **they can become good at computing**.
- More students saw how computing is **useful**.
- 100% of the students **liked the course**.
- 100% of the students **had fun** during the course.
- 100% of the students said that the course made them want to **know more about computing**.
- 80% of the students said that course made them **more interested** in computing.
- 90% of the students said that they **learned more** about computing because of this course.
- 80% of the students said that they **will use** what they have learned in the course.
- 80% of the students said that they learned how **computing can be used** in different ways

through this course.

- Their **favorite parts** of the course were the **activities planned** for each of the classes.
- The **least favorite** part of the course were the **malfunctions** involved with the robot and that was the most popular answer for **changes that they would like** in the course.

There were also some negative results which included the following:

- **Fewer** students wanted to take computing classes involving robots. This may have reflected their frustrations caused by the **malfunctions** of the robots.
- **Fewer** students strongly agreed that they liked the **challenge of computing**.
- At the end of the course, 40% disagreed that they learned more about jobs and only 30% agreed.

Overall, our approach seems successful: our course was well designed and succeeded at getting the concepts across. However, a more robust dataset is needed to further improve the course and observe its effects.

Future Work

Our next step is to adapt the course based on the feedback from the students and our observations. In addition, the course needs to be implemented among a more diverse group of students. In the near future, we hope to build an interactive interface in which animation will complement the text. The interface will provide students with instant feedback to help strengthen their understanding of computing concepts.