

# Young Minds Storming Through Challenging Computer Science Concepts<sup>1</sup>

Anissa Agah St.Pierre, Sarah Carruthers, Yvonne Coady, Rebeca Dunn-Krahn,  
Sophia Dunn-Krahn, Celina Gibbs, Georgia Gibbs, Haley Gibbs, Steven Lonergan,  
Jeff Proctor, Ulrike Stege<sup>2</sup>, Ciara Storey, Margaret-Anne Storey  
Department of Computer Science  
University of Victoria, BC, Canada

## Abstract

*Although the number of software jobs in industry never declined, it is a North-America wide phenomenon that the number of Computer Science students is far below its capacity. One of the challenges in promoting Computer Science to young students is communicating how it can be engaging and relevant for them. As a result, many children often lose interest in Math and Computer Science before they even get started. Over the past year, a team of faculty and students from the University of Victoria have participated in outreach efforts to over 500 children across British Columbia. These efforts have successfully engaged elementary school children in challenging Computer Science concepts through robotic-programming missions. The feedback from these activities has been overwhelmingly positive. This paper shares the design and deployment of these outreach efforts and discusses the positive impact they are already having. We conclude by proposing an expanded set of outreach activities that can be conducted by institutions across BC and Canada in a coordinated fashion.*

## Introduction

It is ironic that, though children today are growing up immersed in technology, surrounded by and connected through ubiquitous devices, “computer science is in perpetual danger of disappearing from schools” [1]. Building on the success of the Eliza Kuttner’s Programming Camps for Girls [2], the goal of our project is to actively communicate the wide diversity and applications of Computer Science to grade school students. We aim to banish the stereotypes that are pervasive in today’s media and attract participation from non-traditional groups. To this end, we identify major stepping stones along the way to include the introduction of Computer Science as a mandatory subject in all primary and secondary schools in BC, and a change in the public mindset about Computer Science. Though major obstacles include achieving sufficient buy-in from stakeholders such as the Ministry of Education and teachers, this paper focuses on the stakeholders we have been in closest contact with over the past year students from grade 3 to 12. Our outreach efforts indicate that not only are these students capable of working with challenging Computer Science concepts, but they are willing and eager to do so. Based on our experience to date, we propose a cross-institutional plan to investigate how early Computer Science concepts, problem solving and logic skills can be introduced through fun and engaging hands-on activities.\

## Current Efforts: Design and Deployment of Mindstorms Missions

Our activities today centre on the notion that play nourishes every aspect of children’s development. Play forms the foundation of intellectual, social, physical, and emotional skills necessary for success in school and in life. Play “paves the way for learning” [9]. Given a fun and interactive learning environment, we believe that many core concepts that are part of a basic Computer Science education can be taught to

---

<sup>1</sup> We thank the Natural Science and Engineering Research Council Canada (NSERC) for an NSERC Pacific Discretionary Fund Award supporting the "LEARN! Logic and Engineering Adventures in Robotics, Now!" project.

<sup>2</sup> Corresponding author: Ulrike Stege, Department of Computer Science, University of Victoria, PO Box 3055, STN CSC, Victoria, BC, Canada V8W 3P6, phone: +1 250 472 5729, fax: +1 250 472 5708, email: [stege@cs.uvic.ca](mailto:stege@cs.uvic.ca)

young learners. These concepts include algorithm, data structure, input, output, time and space complexity, conditional branching, looping, recursion, modularity, file I/O, security, encryption, communication, binary numbers, random numbers, polling, variables, and concurrent programming. In their activity book “Computer Science Unplugged”, Tim Bell, Ian Witten and Mike Fellows demonstrate how to explain notions such as binary numbers, sorting algorithms and data structures as in-class activities and without the direct use of technology [8]. Early outreach activities in Victoria successfully used these techniques to get very young students interested in the fundamental topics underlying Computer Science.

Recently, we have extended these unplugged activities, by teaching core concepts using the LEGO Mindstorms Robotics kits. University students and professors at the University of Victoria launched the Lego Robotics Festival using the latest Lego Mindstorms NXT robots [3]. Grade 6 to 12 girls from across Greater Victoria were invited to learn fundamental engineering and programming concepts through hands-on experience with technology while exercising problem solving, team building and interpersonal skills. The challenges posed through these workshops are designed and led by Computer Science university students.

The teams of 5-6 girls employ key elements of Extreme Programming [4] to iteratively build a robot, design a problem solution, program the robot using a programming environment on a computer and test their solution in pairs. Concepts taught in the LEGO Mindstorms workshops include algorithms, conditional branching, looping, modularity, input, output, file I/O, random number generation, concurrent programming, polling and variables. Through these interactive and collaborative workshops, students develop further essential skills, such as problem solving, algorithmic thinking, communication and team-work skills, and engineering skills. The youth also become familiar with basic mechanical and electrical engineering concepts, such as construction of a balanced vehicle, use of gears, application of sensors, basic electronic circuitry, and concepts of speed versus control. Future workshops to teach concepts of security, encryption, and communication are currently being developed.

Beyond acquiring social skills and further developing engineering and scientific knowledge, students also have the opportunity to be creative as they build the robots and solve the tasks in a tangible, hands-on manner. The skills acquired through these workshops will benefit the students throughout their lives, whether they continue in the sciences, engineering, arts or education disciplines.

While these activities introduce young minds to programming in a fun and familiar environment, the content of the workshops is strategically planned to teach core Computer Science concepts in a non-invasive way.



**Figure 1.** “Ciara Storey, left, cheers when her team’s robot stops where it’s supposed to during a robotics workshop at the University of Victoria on Saturday. Storey and her team-mates: Cathleen Evans, Christine Scheurle and Amber Robinson, named their robot Susie-and-the-Cream-Cheese-mobile.”[5]

The LEGO Mindstorms NXT kits that are used in these workshops consist of a central block (the NXT), a set of sensors (touch, ultra sonic, sound and light), 3 servomotors, and a wide selection of LEGO building pieces (including gears, wheels and structural pieces). The

NXT contains a 32-bit ARM7 microcontroller, 3 output ports (for motors), 4 input ports (for sensors), a USB port, a loudspeaker, 4 buttons, a display and Bluetooth wireless communication. Internal memory is used to store programs, sound files, graphics files and text files. Sensors can be mounted in a myriad of different configurations, with up to four connected to the NXT at a time. The NXT supports up to three motors, connected in various fashions, and powered by three dedicated motor ports. Bluetooth capability allows communication between NXT’s and other Bluetooth-enabled devices. The versatility of this product makes the Mindstorms a powerful teaching tool, offering endless opportunities for creativity and design.

Programs are currently created for the NXT using the Mindstorms NXT software, a drag and drop development environment, and then uploaded to the block via USB or Bluetooth. There are other development environments available as well, which can be used in the future to expand the reach of this program. The Mindstorms software is open source enabling the development of custom firmware and development environments. It could also be used to teach more advanced concepts in later workshops, letting students work on the building blocks themselves.

In a typical daylong workshop, the students spend the majority of the morning building a robot for the mission using a set of instructions. Much room is left for creativity once they have completed the basic robot. Later in the morning, the instructors walk the students through a series of progressively more challenging programming modules. The modules are designed to cover a specific subset of core concepts and skills in Computer Science, while simultaneously introducing the students to the LEGO Mindstorms hardware and programming environment. Once the students have successfully completed all the morning modules, they are given a challenge problem, which they tackle in the afternoon portion of the workshop. The teams of 5 or 6 youth design, implement and test a solution to the given problem using skills and concepts they have learned during the morning.

For example, in the Hovercraft Rescue Mission, students begin by building what is known in the LEGO Mindstorms community as the Tribot: a simple robot with two motorized wheels and a third, non-motorized, pivoting wheel at the rear of the Tribot. They are given simple, step by step instructions, including how to properly connect the motors to their respective ports, and how to mount and connect sensors. During the morning portion of the workshop, students work their way through modules designed to cover a specific set of core concepts: conditional branching, looping, input, output, polling, modularity, reading and writing to variables, and data type conversions. Once all these topics are adequately covered, students are given their final mission. They must program their already constructed Tribot to drive along a river to reach an unknown number of stranded people (simulated by marbles). They can design a container for the Tribot to transport the people one by one, or several marbles at once safely across the raging river to safety. They are required to carry out the mission with at least two trips to ensure that they use looping to solve their challenge.

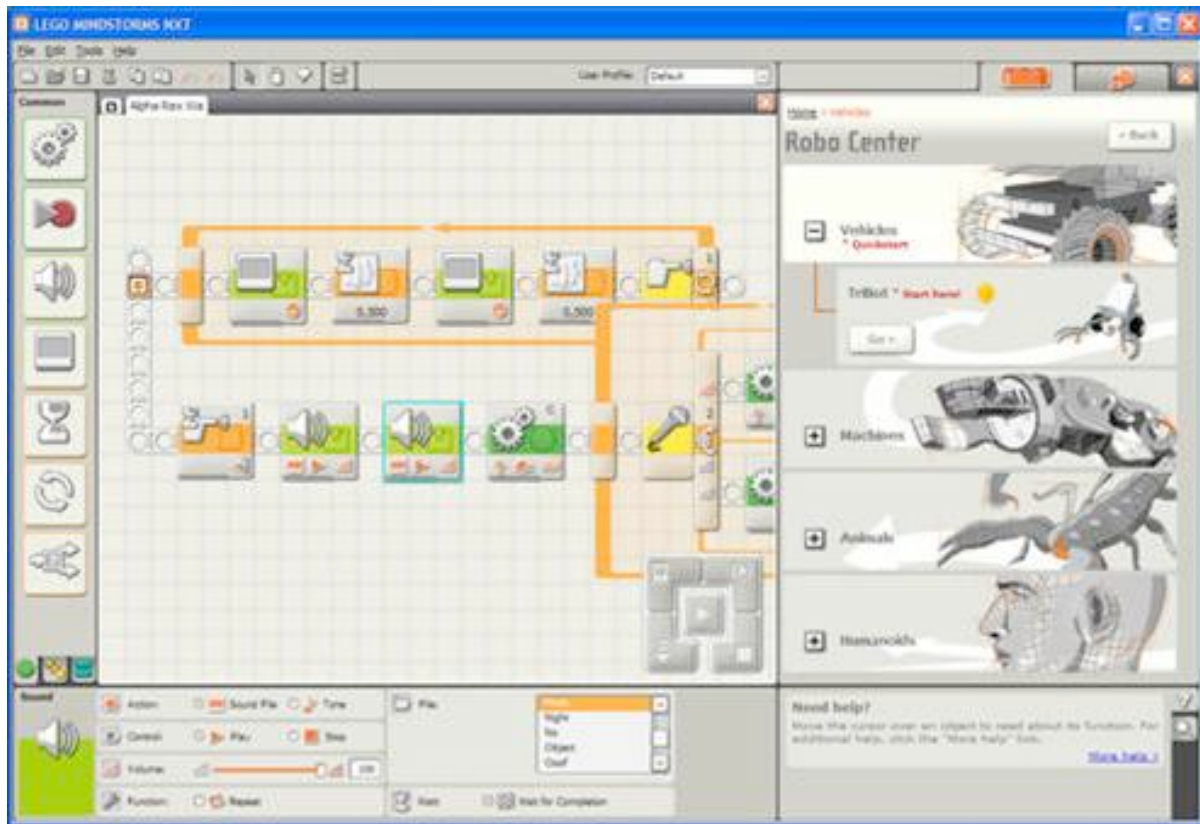
The Hovercraft Rescue Mission challenge is presented to the students with a set of very clearly laid out parameters and limitations. Students are given boundaries to work within: the physical set up of the simulated river, instructions detailing how many passengers may be transported at a time and safety guidelines for the simulated people. This structure is appealing to beginners and some types of learners, providing them with easy-to-understand objectives and guidelines against which they can measure their progress.

To appeal to other types of learners, or those who want more freedom in their learning environment, our team developed the Dance Workshop. While it still focuses on presenting core concepts in Computer Science, the Dance Workshop requires and encourages more creativity. Students spend the morning covering the following concepts: looping, input, output, modularity, variables, random numbers, conditional branching, Boolean logic, and concurrent execution. For their afternoon challenge mission they are asked to design a robot that can dance along to music, with movements that are controlled by both the level and rhythm of the music. They are asked to add a mechanical “head” to the Tribot, which should move to the music. The creativity required for this mission is particularly appealing to girls. By presenting a challenge that requires creativity and artistic flare, we hope to dispel the misconception that Computer Science is boring and mundane, and show students that there is considerable room in the field for creative thinkers.

There are a number of reasons why the Mindstorms workshops have been so successful. First, most students are already familiar with LEGO products. All of the LEGO products are designed with young people in mind: the building instructions for the Mindstorms are elegant in their simplicity. Since they are graphical, they do not require students to read written instructions. Similarly, the development environment is graphical in nature: students drag and drop blocks onto a “sequence beam.” The blocks have simple icons on them indicating their nature, and the flow of the program is left to right along the sequence beam.

The simplicity and design of the LEGO Mindstorms makes it possible to use them to teach young people, but a more important advantage is that the LEGO Mindstorms are fun and hands-on. In our opinion, this is key for piquing the interest of young learners. For example, when teaching students I/O and the use of sensors, we avoid the typical theoretical approach used in most classroom environments. Instead, students are given practical examples they can relate to in their daily environment. Students are encouraged to give examples of programmable items they use regularly and identify what is the input and what is output for those objects. Examples include: the phone rings to inform us a call is coming in; a timer beeps to let us know an interval of time has elapsed; we press a button on a computer to turn it on; we type information into a form online. Other examples used to relay the concepts of input/output and sensors are biological: for instance when we hear someone speak, we receive input through our ears; when we speak, our voice is an output. Our brain processes and makes a decision to create an output (or not), based on the input. The computer can be viewed as a model that mimics our brain’s ability to make decisions based on the input it receives and then provide appropriate responses using an output capability.

With this new understanding and realization, students program their robot to receive input and generate output in a sequential manner. Our young learners then download their code onto the robot and test it to determine if the robot is behaving as anticipated. The problem solving activities continue as they test and debug their program with the help of university students.



**Figure 2.** Snapshot of the Lego Mindstorms NXT Software v1.0 graphical programming environment. [3]

## Impact

So far, this project has been very successful at promoting awareness of computer science to underrepresented groups. Students quickly pick up core concepts and begin applying them in their own solutions. We hope this introduction may lead to higher enrolment levels in post-secondary computer science programs in the future. Students that choose to enter a post-secondary computer science program will benefit from an introduction to fundamental concepts in the field before starting their first year in University. For all students, this program encourages the types of critical thinking, problem solving and logic that are fundamental to many disciplines.

The success of the initial workshops caught the attention of the local media in Victoria as well as by ACM Tech News. A CBC radio interview was held on February 23, 2007 with some of the participating elementary school students. When asked in a radio interview what they liked about the robots, one participant related her experience on her success as “I can actually do it” and further explained, “It makes me feel smart”. What was unexpected, however, was the observed activity level and enthusiasm, especially while students were working on their final challenge problem. As they got closer to a working solution, the students ran excitedly back and forth between the laptop (where they modify the program), and the test location (typically an open space on the floor). With each iteration, their energy level rose. In fact, two young participants (aged 8 and 10) that worked with an earlier version of the mission identified the testing phase as their favourite part of the activity. It is easy for them to gauge the correctness of their solution, and therefore easy for them to make necessary adjustments or corrections. This gives students a real sense of ownership and builds their confidence. This kind of response is what we are hoping to achieve with this project, that is, to naturally introduce computer science as an option for

discovery in their future education, while simultaneously increasing their feelings of self-worth.

An additional impact, that was not initially expected, has been the community that has formed around the goals of improving the perception of and increasing the enrollment in computer science. The community involves enthusiastic children, university students, graduate students and faculty. These individuals are working together to expand the outreach both within and beyond the discipline.

## **Future work**

So far our efforts have been successful in introducing Computer Science concepts to girls in Grades 6 to 12 in the Greater Victoria area. However, we plan to reach and challenge a much broader audience. We outline possible expansions of the project. There are a number of ways in which we can expand upon the work we've already done with the Mindstorms workshops. We can use them as a building block to develop engaging workshops for first year university students. The workshops can also easily be extended to teach more advanced concepts to young learners by modifying their format. Other products are worthy of investigation, like Pico Crickets [6], are available, and have been successfully used elsewhere to promote Computer Science awareness. Finally, one drawback of the Mindstorms workshops that we need to address is that they are not readily accessible to many young students in the province due to the cost of the equipment required and the need for some programming experience to run them. We plan on addressing this shortcoming through the development of programs that make use of low tech lessons and free downloadable software tools.

First year computer science courses often introduce programming in textual languages like Java and C. These languages may not be ideal for introducing young students to the world of programming and yet the topics taught in first year are many of the same topics taught to younger students using graphical programming such as that offered by the Lego Mindstorms NXT. We wish to explore if the Lego workshop activities can also provide some interesting stepping-stones for first year university students. There are also a number of tools to write programs for the Lego Mindstorms NXT in commonly used textual languages, thus providing a mechanism for transitioning first year students from graphical programming to traditional text based languages.

Building upon the positive experience of the LEGO Mindstorms projects, we are planning to offer more advanced workshops, where students can further pursue studies in Computer Science using interdisciplinary applications. The University of Victoria currently has several interdisciplinary programs in computer science. Workshops that pick up on these themes (e.g. robotic instruments) and broaden the applications explored will further enhance the attractiveness of the field to other students and youth, while also leading to more exciting workshops.

We are also currently planning to include in our workshops other products that can be used in a similar way to teach core concepts in Computer Science. Such an example is the Pico Crickets from MIT's Media Lab [6]. Pico Crickets use a similar feature-rich, visual programming interface as LEGO Mindstorms but allow students to explore a broader range of open-ended projects. Recently MIT set up an event titled "A Day in the Park" [8] which invited people to create objects they would find in a park. At the end of the day participants displayed the objects in a pre-constructed park setting. These creatures, objects, and people were capable of moving, lighting up, and even detecting their surroundings using various Pico Crickets sensors. Scratch [10], a programming language based on Pico Crickets, does not need the Pico Crickets base to run its programs. It uses colour schemes and block shapes to help students to grasp concepts, and eliminates the need to read compiler feedback that can be baffling to a new programmer. Scratch doubles as not only an effective way to program when certain resources are not available, but also as something students can take home after a session using Pico Crickets. The similarity of the interface means that it is easy for students to continue to program at home (or school) with Scratch after a workshop, allowing workshops to reach out to the students long after initial contact.

We recognize that the workshops we and other institutions plan to offer may not be sufficient. Therefore we propose to have lesson plans<sup>3</sup> and workshops available to teachers outside the Victoria area – in reaching the teachers. We will explore solutions that do not involve the financial burden associated with purchasing Lego Mindstorms. This strategy includes using free downloadable software and web applications that will not only be available to students during the workshops or lessons, but also at home if they wish to continue learning.

Also in development are a series of MAPS (Math and Problems Solving) activity bundles, which teachers can present in class at a very low cost and with little or no specialized training. The MAPS activity packages will consist of a series of completely self-contained bundles, containing all the resources (except for commonly found supplies: string, paper, sticks, etc.) and information necessary to run a specific problem solving activity. Each bundle will focus on a specific subject of interest and/or core skill in Computer Science, and will be designed for presentation by a teacher to a group of up to 30 students. These will not be just sit-down, pencil-and-paper activities, but rather activities or games that can be done outdoors, in a gymnasium, or in the classroom.

A key future effort will lie in the experimental evaluation of our efforts to measure the successfulness of the proposed activities. Possible parameters may include the amount of interest (in number of students) as well as increased exposure to computer science applications. An effective evaluation can likely only be done following a coordinated effort across BC to conduct outreach activities to improve enrolment in computer science.

## **Conclusion**

Interest in computer science remains a major obstacle in the recruitment of new students. A stigmatic conception of programming in general deters many kids. Computer science is seen as boring, difficult, and inaccessible. The Lego Mindstorms workshops have been effective in stimulating interest in programming concepts. They are successful in making computer science fun and interesting, while at the same time, challenging kids to think about meaningful programming topics. More advanced textual-based or open-ended programming environments could extend this success to a tangible understanding of basic programming. It could be valuable to offer kids a perspective of programming languages in practical use. The slow progression from high-level to low-level concepts would help to demystify intimidating impressions of computer science.

The Lego workshop serves as an excellent introduction to computer science for kids. It provides a high-level perspective of programming concepts. The robots serve as a bridge into otherwise foreign computer science material. They provoke interest by making programming fun, and stimulate learning by presenting real computer science topics. However, it is still a big leap to picture a program graphically, and to design a textual program. New programming environments could help usher kids into post-secondary computer science education. As well, the cost and knowledge required to offer these workshops excludes a huge portion of potential learners the British Columbia. They are simply not available to students who live in smaller or remote communities. These options provide a way to reach a broader range of students than those just in Victoria. Furthermore, because of the use of features like free tutorials, low-tech activities, and free software, teachers can easily offer these activities and workshops without training. This format also allows the students to continue to learn, play and create long after their one-day workshop.

Currently, these LEGO Mindstorms workshops are being offered to girls in grades 6 to 12. In the near

---

<sup>3</sup> For an already existing example of lesson plans see [8].

future, our plan is to expand this audience to include both boys and girls, and to develop different types of programs. In particular, multi-day and week-long workshops would allow students to get a more in depth experience on a specific area of study, or allow for a brief introduction to an overview of Computer Science. These longer programs could include sessions of more formal theory, active games and outdoor activities, field trips, and a more challenging final project.

## References

- [1] A. Tucker. A model curriculum for K-12 computer science: final report of the ACM K-12 Task Force Curriculum Committee, 2<sup>nd</sup> edition, 2006.
- [2] Eliza Kuttner's Programming Camps for Girls, <http://www.cs.ubc.ca/~pcarter/bridgesProject/index.html>.
- [3] Lego Mindstorms NXT, <http://mindstorms.lego.com/>
- [4] Extreme Programming, <http://www.extremeprogramming.org/>.
- [5] Sharon Tiffin (Oak Bay News). "Girls take to robotics", Saanich News. Feb 21, 2007.
- [6] Pico Crickets, <http://www.picocricket.com/>.
- [7] N. Rusk, M. Resnick, R. Berg, M. Pezalla-Granlund (In preparation for submission). New Pathways into Robotics: Strategies for Broadening Participation, <http://web.media.mit.edu/~mres/papers.html>.
- [8] T. Bell, H. Witten, and M. Fellows. Computer Science Unplugged, December 2006.
- [9] <http://www.ccl-cca.ca/CCL/Reports/LessonsInLearning/LinL20061010LearninPlay.htm>
- [10] Scratch, <http://scratch.mit.edu/>.