Comparison of an Introductory Engineering Course with and without LEGO Mindstorms Robots

Harold N. Scholz, Pennsylvania State University Lehigh Valley; Kevin McFall, Pennsylvania State University Lehigh Valley

Abstract

An introductory engineering course was run in two concurrent terms, one without LEGO Mindstorms robots, and one with. A comparison was performed between the two approaches considering teamwork, leadership, and engineering problem solving. In both groups, the project covered the engineering design process from customer needs through a working prototype.

The first semester, student design teams were allowed to choose any toy-related project. In the second, they were required to use the LEGO robots to navigate a maze, find a colored ball, and return it to the starting point. During the non-robotic semester, one or two members of each team tended to dominate the group, with some members seldom contributing. With the robotics groups, leadership changed throughout the engineering process as expertise of different individuals became important. The students were involved throughout the project as prototypes did not work and both mechanical and software changes were required. The robotics project required not just mechanical expertise, but also the ability to program. The LEGO system also introduced many of the students to programming for the first time through a graphical interface that allowed everyone to participate.

Introduction

An engineering class designed for incoming freshmen has been modified to be project-based. The class, Introduction to Engineering Design, was administered in two consecutive semesters at Penn State University Lehigh Valley during the 2009–2010 academic year.

The goal of the course is to expose students to the engineering design process, methods, and decision-making. A team-based approach was adopted with grades based on team presentations, written journals, and a successful project. The course covers the design process in detail from customer needs to a working prototype as shown in Figure 1. Students are required to generate presentations for customer needs, product specifications, concepts, and intermediate and final prototypes. The scope of projects during the first semester was virtually unlimited. The only restrictions were that the project be a toy and that it be approved as realistic. One of the potential approaches was to use the LEGO Mindstorms robots (http://mindstorms.lego.com/en-us/default.asp).

During the second semester, projects were restricted to generating a robot to navigate a maze, find a red ball within the maze, and bring it back to the starting point. The robots also had to ignore any blue balls along the way. Each team was still required to define the form of the robot, method of movement, etc., and incorporate customer needs.

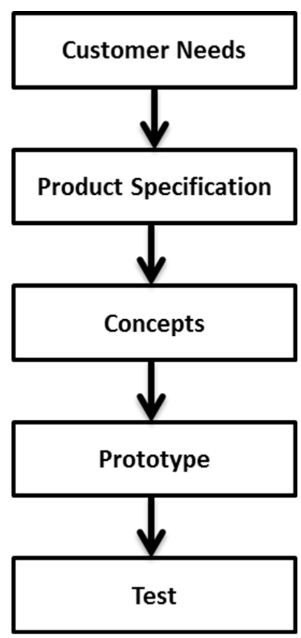

Figure 1: Design Process

Semester 1 Observations

During the first semester, each team started with four members. Teams were generated based on the skill sets of the members of the class. All students were administered a survey to assess their mechanical, electrical, and programming abilities [1]. Based on the survey results, teams were selected in an attempt to equilibrate skills across teams. Each team was instructed to build a prototype of one toy. They had to decide what toy they wanted to design and obtain input on customer needs to determine the right product. Requiring students to define the product proved to be an important step, as they were asked to narrow a broad spectrum of choices. The students used resources such as the Internet, visiting elementary school classes, and talking to people they knew. This portion of the project provided a good challenge to the students and produced relatively good results.

Generating product specifications forced the teams to define all aspects of the selected toy. This activity held the attention of the majority of students and produced good results. The concept phase was also successful in terms of results and participation of most of the students. However, as this phase unfolded, one or two students in each group began to take over as leaders. In each case, the leaders pushed the projects in a direction where they had some expertise. In the prototyping stages, the same one or two team leaders continued dominating the project in two-thirds of the teams. While two of those teams did produce prototypes similar to their concepts, none of them worked as planned. One team did produce a working prototype, but it was lacking the full function prescribed by the concept.

The group that decided to do a LEGO robot project also fell short of their goals, but the dynamics of that group changed from the start to the final prototype. In the beginning, one group member was leading the group. Once the construction of the robot was completed, the leadership shifted to another member of the group during the programming phase. All group members of the robotic project were highly involved in at least some portion of the prototyping phase.

Semester 2 Observations

Teams were generated in a similar manner as in the preceding semester. When teams were divided, the first priority was programming experience. Next priorities were mechanical and other general skills. Each team was instructed that they were to build a toy that had a specific task to perform. It was to navigate a maze and capture a red ball, bringing it back to the start. This had to be accomplished using the LEGO robotics set provided. Teams were also allowed to build additional parts if required.

During the customer needs phase, students had some difficulty compared with the previous semester, in which free reign was given over what to build. In this case, teams were not nearly as imaginative and did not do as well at surveying potential customers. The product specification phase was also weaker during the second semester, possibly due to the lack of excitement of building ideas of their own. Two of the six groups had reasonable product specifications on the first attempt, but the other four did not.

Once the specifications were completed, teams were introduced to the LEGO sets. At this point, they began developing their concepts. The concepts did follow ideas obtained from customer needs and developed relatively well. Some of the ideas generated were concerned with locomotion; for example, tracks vs. wheels and how to configure them. Other ideas dealt with how to capture the ball. All the groups failed to document concepts on how to program the robot.

Another interesting learning situation occurred once students were given the LEGO sets. Many teams began building prototypes as opposed to generating as many concepts as possible. This did help them quickly narrow down the final concept that they presented, but most made fairly major changes to the designs as they progressed into the testing phase. The presentations for the concept phase were strong. Students could visualize what they were going to design/build and had pictures as well as text in their slides. During the concept phase, one or two leaders of each group essentially controlled the direction.

During the prototyping phase, the same leaders of each group led during the mechanical build phase. In two-thirds of the groups, the leadership changed once the robotics programming began. Students that had been fairly reserved and not significant contributors to this point became the leaders of the group. Other team members were still heavily involved, since all groups had unforeseen mechanical issues as well as continued software modifications.

The programming used a simple drag-and-drop graphical interface, as shown in Figure 2, so it was not beyond the abilities of the students. Also demonstrated in Figure 2 is a continuous loop with a conditional statement that turns left if the button is pushed, otherwise it turns right. This program eliminated the need to learn sophisticated syntax and students could focus on the problems and algorithms. This benefit of graphical programming has been reported by multiple authors [2-4]. On several occasions, the main programmer was missing for a portion or all of a class. Teammates would pick up and make updates to the programs. In some cases, these team members had no prior programming experience.

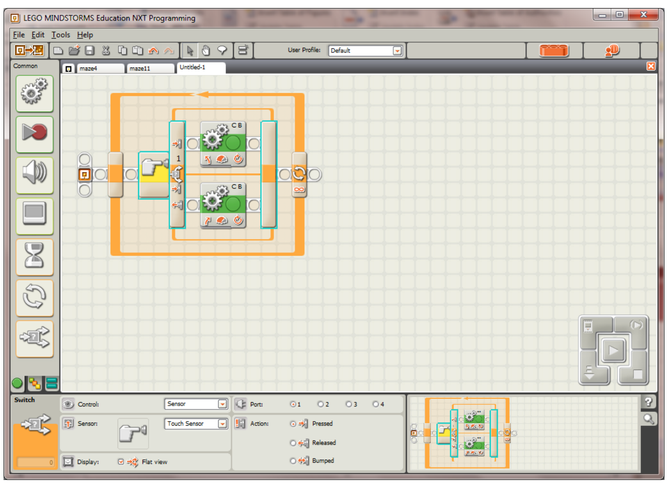

Figure 2. Graphical Drag-and-Drop Programming Interface and Program

Size constraints arose as unexpected issues were faced by the groups. The maze was rather constricting compared to the size of the robotics. This caused numerous redesigns for all the groups. They also encountered stability issues where the robots would tip over while cornering or when striking a wall. Also, teams realized that the "simple" mechanism originally planned for capturing the ball was not as easy as it first appeared.

The teams met with multiple challenges in developing the software. A simple gap in the outer wall of the maze would cause robots to turn into the wall. Navigating in a straight line would not work as planned, since the motors of both wheels did not always turn at the same rate. Most of these issues required including feedback or damping the reaction to a sensor change. Besides learning engineering skills during this phase, students learned to work together as a team. Each team member had different skills to bring to the project at different points. Also, the time to build and program the robots was short enough to instill a real sense of urgency.

At the end of the project, one group successfully navigated the maze and captured the red ball. All of the groups were able to navigate the maze with varying degrees of sophistication. A better measure of success, however, was that all groups worked together as teams and made good progress towards a goal. The last day of class, the maze and robots were on display in a central area of the campus where the final projects were demonstrated for students, faculty, and staff.

Comparisons

Allowing students the freedom to choose any project definitely produced more interest during the initial phases of the project. Through the concept phase, non-robotics students were more enthusiastic and creative than students with a robotic project. However, the non-robotics groups got bogged down during the prototyping stages, and all but one group seemed to lose track of what they were trying to accomplish. Contributing to the confusion was the fact that the goals set out when defining product specifications were more difficult than envisioned. As freshmen, they did not have the appropriate skills to actually complete the projects as specified. On the other hand, the robotics students showed increased enthusiasm as they started working with the LEGO robots during the concept development phase.

The robotics teams knew it was possible to build a robot to do what was being asked of them, and they could see other groups getting closer to the solution. Since all teams had the same raw materials available, it put them on equal footing. This definitely helped them to focus and work hard to get their projects to function. Participation in the non-robotics group stagnated, with one to two people from each group leading from start to finish. The robotics groups, however, saw at least three of the four members take control at some time in all but one of the groups.

Conclusion

Use of LEGO robotics provided an attainable engineering challenge for freshman engineering students. They were reasonably engaged during the customer needs and product specifications phases. During the concepts and prototyping phases, the vast majority of students were heavily engaged in solving the many problems associated with the project. Many more students were involved in leading their respective teams with the robotics than with the non-robotics projects. This was due to the variety of work that had to be accomplished in a relatively short time.

Acknowledgments

The Toys 'n More project at Penn State University provided funds and inspiration for this project. The Toys 'n More project is supported by NSF grant 0756992.

References

[1] Case, Carl J. "A Systems Model for Designing and Implementing the Technology-Based Team Project Course." *Proceedings of the Academy of Information and Management Sciences, Volume 5, Number 1.* Nashville, 2001. 16-19.

[2] Carlisle, Martin C., Terry A. Wilson, Jeffrey W. Humphries, and Steven M. Hadfield. "RAPTOR: Introduction Programming to Non-majors with Flowcharts." *Consortium for Computing Sciences in Colleges.* Central Plains, 2004. 52-60.

[3] Crews, Thad, and Uta Ziegler. "The flowchart Interpreter for Introductory Programming Courses." *Proceedings of FIE'98.* 1998. 307-312.

[4] Nandikolla, Vidya K., and Suhas S. Pharkute. "Flowchart Visual Programming in Mechatronics Course." *ASME Early Career Technical Journal, Volume 9*, 2010: 118123.

Biographies

HAROLD SCHOLZ earned his BS in Natural Science and MS in Physics from the University of Akron in 1975 and 1976, and his PhD in Physics from The Ohio State University in 1981. He is currently an instructor in Physics and Engineering at Penn State University Lehigh Valley. Dr. Scholz has more than 26 years of experience in integrated circuit design and research and development at Bell Laboratories, Lucent Technologies, Agere Systems, and Lattice Semiconductor as both an engineer and manager designing integrated circuits. Dr. Scholz may be reached at hns12@psu.ed.

KEVIN MCFALL serves as an assistant professor at Penn State University Lehigh Valley teaching a range of undergraduate engineering courses. His research program involves applying computational science to solve engineering problems using artificial intelligence techniques. Dr. McFall earned bachelor's, master's, and doctoral degrees in Mechanical Engineering from Virginia Tech, MIT, and Georgia Tech, respectively. He may be reached at ksm21@psu.ed.