Engineering Computer Games: A Parallel Learning Opportunity for Undergraduate Engineering and Primary (K-5) Students

Mark Michael BUDNIK
Heather Ann BUDNIK
Electrical and Computer Engineering, Valparaiso University
Valparaiso, IN, 46383, USA

ABSTRACT

In this paper, we present how our College of Engineering is developing a growing portfolio of engineering computer games as a parallel learning opportunity for undergraduate engineering and primary (grade K-5) students. Around the world, many schools provide secondary students (grade 6-12) with opportunities to pursue pre-engineering classes. However, by the time students reach this age, many of them have already determined their educational goals and preferred careers. Our College of Engineering is developing resources to provide primary students, still in their educational formative years, with opportunities to learn more about engineering. One of these resources is a library of engineering games targeted to the primary student population. The games are designed by sophomore students in our College of Engineering. During their Introduction to Computational Techniques course, the students use the LabVIEW environment to develop the games. This software provides a wealth of design resources for the novice programmer; using it to develop the games strengthens the undergraduates' programming skills and reinforces their early engineering lessons. The games then provide an opportunity to introduce next generation to basic engineering concepts.

Keywords: Engineering, Programming, K-5, LabVIEW, Pre-engineering.

1. INTRODUCTION

There are few professions that are shrouded in more mystery than engineering. When questioned about the role engineers play in society, most people will answer with generic comments such as "They build things," or "They design stuff." The specific challenges faced by engineers are much more daunting than just building or designing. Each day, engineers must quickly find optimal solutions to problems while using the least possible amounts of time and other resources. Engineers will play an increasing role in mankind's ability to thrive in the 21st century as our global resources continue to be stretched.

Our global economy needs the skills of engineers more than ever before, but the general population continues to be confused about what engineers actually do. Most secondary school students and most engineering freshman would find it difficult to describe an engineer's job. Throughout the country, many schools are giving more attention to pre-engineering education. However, most of their focus is on secondary students, and many of these individuals have already decided on educational goals and career paths. Few resources have been dedicated to the elementary (grades K-5) student population, where we have a tremendous opportunity to excite children about the prospects of a career in engineering [1-2]. A survey of the past four years of the *Journal of Engineering Education* (published by the American Society for Engineering Education) finds a number of papers addressing pre-engineering in K-12 education [3-8]. However, none of these papers focus on elementary pre-engineering education. Some consider the topic briefly, but most do not mention it at all.

2. HYPOTHESIS

Our College of Engineering is working to revitalize the development of pre-engineering resources that are appropriate for elementary school students [9]. We plan to develop a new array of short computer games intended to teach young students introductory engineering concepts while reinforcing young undergraduate students' knowledge of engineering fundamentals and computer programming [10]. The games are intended to be fun but will always include an application focus to help the elementary and undergraduate students better understand the role and work of engineers in the 21st century.

3. UNDERGRADUATE ENGINEERING GAME DEVELOPERS

To best utilize limited resources, we offered our undergraduate engineering students the opportunity to help develop the games. Their own engineering abilities will be sharpened by reinforcing the skills that they have learned in their engineering and programming classes.

The undergraduate engineering game developers were all students in an electrical and computer engineering (ECE) class: ECE200 - Introduction to Computational Techniques [11]. This class is traditionally taken by third semester students at Valparaiso University. The class is taught in a combined lecture and laboratory format, with approximately twenty-five students per section. Each laboratory classroom has sixteen computers, so students generally have their choice of working individually or with a partner.

The ECE200 class follows a two-semester course sequence of GE100 and ECE110 (Introduction to Engineering and Introduction to Electrical and Computer Engineering, respectively), which are typically taken during an engineering student's freshman year [11]. In these classes, the engineers are taught basic principles of electrical and computer engineering in order to introduce them to the field. These ideas are further developed in a linear circuit class and a digital logic class, also taken in the third semester in parallel with ECE200.

The intention of ECE200 is to introduce our students to various computer tools that can be used to solve electrical and computer engineering problems. Lessons are taught in PSpice (an electronic circuit simulator) and LabVIEW (a graphical programming language). One of the five objectives of the class is for students to be able to develop new graphical programs using LabVIEW. Therefore, the decision to have the students develop the engineering game in LabVIEW as a final project for ECE200 was a natural fit. Developing the games gives the ECE200 students a chance to further develop and refine their LabVIEW programming skills and reinforces the engineering lessons they have learned in other classes.

4. PROGRAMMING LANGUAGE FOR THE DEVELOPMENT OF THE COMPUTER GAMES

The programming language engineering undergraduates learn in ECE200 is LabVIEW [12]. It was introduced by National Instruments in the 1980s and is widely used in the engineering and science disciplines.

Every LabVIEW program has two facets. The first is the front panel (Figure 1), which displays all of the user's controls (i.e. buttons, dials, numbers, switches, character strings, etc.) and indicators (i.e. gauges, graphs, charts, lights, etc.). The other facet is the block diagram (Figure 2), which is the graphical code that determines what the program does. Data in a LabVIEW program travels down "wires" from user controls, through computer functions, and then to user indicators.

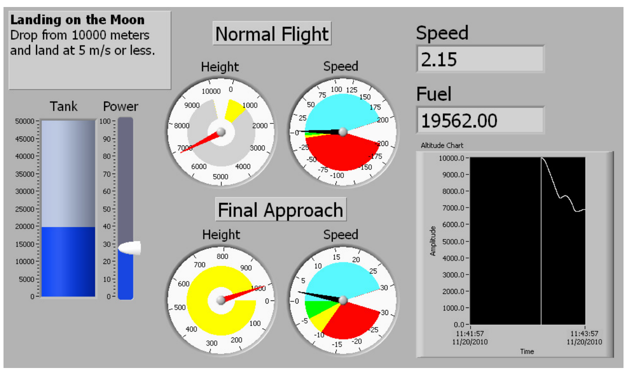

Figure 1. Example of a LabVIEW front panel containing the program's user interface.

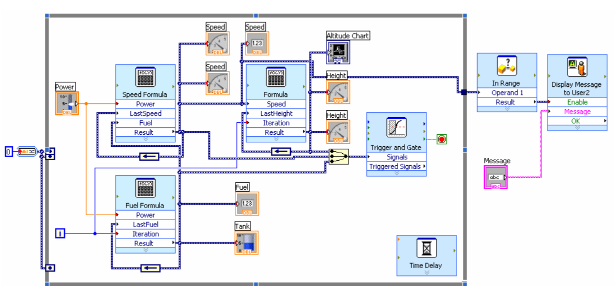

Figure 2. Example of a LabVIEW block diagram containing the graphical code.

Several characteristics of LabVIEW make it an ideal programming language for this project. First, LabVIEW is very flexible. It allows the computer programmers to develop custom user controls and indicators in any size, shape, or format (see Figure 3). Second, the language is very simple to learn - a key feature for a first programming language. A complex program can be developed in a very short time using either library functions or programmer-developed functions (see Figure 4). In addition, the dataflow principles used in LabVIEW make it ideal for students to learn while they are studying analog and digital electronic circuits.

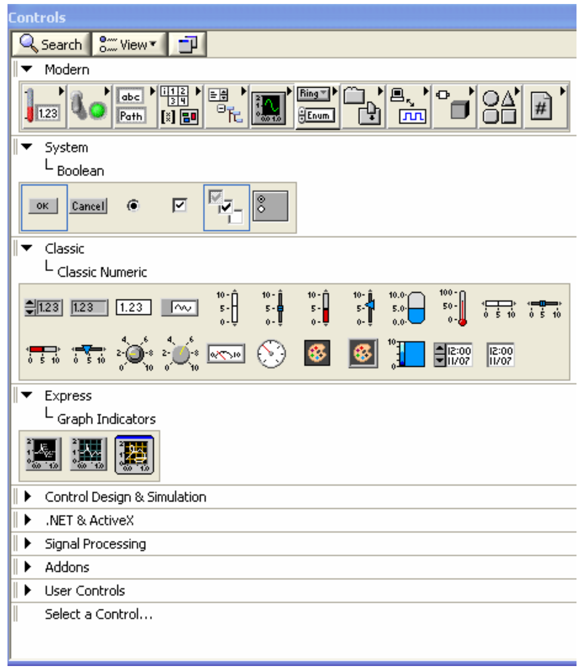

Figure 3: LabVIEW controls palette featuring user numeric, string, boolean, and graph inputs and outputs.

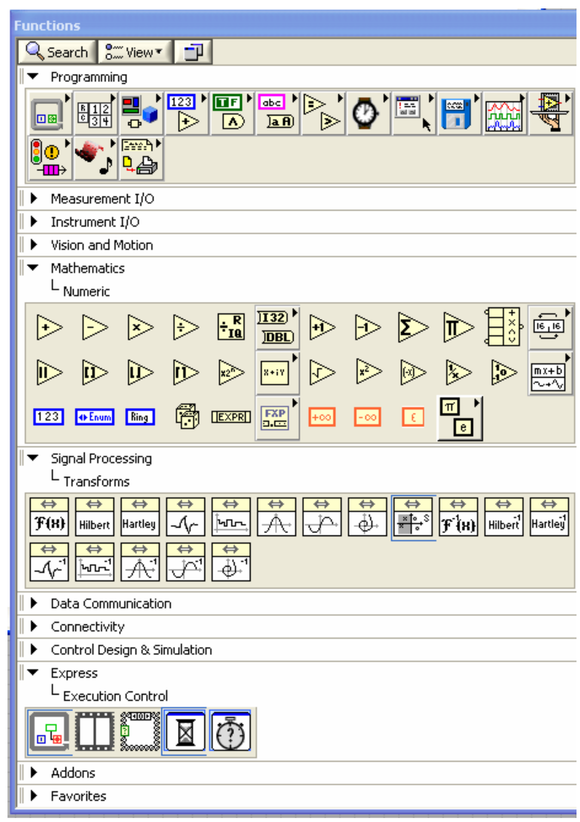

Figure 4: LabVIEW functions palette illustrating some of the numeric, processing, and control functions.

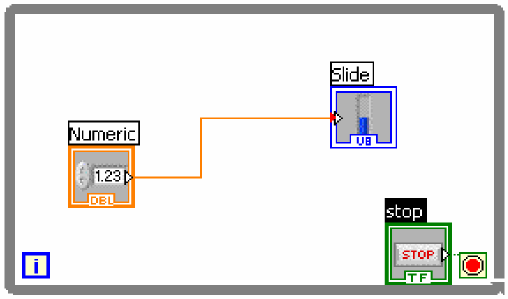

Figure 5: Simple LabVIEW program illustrating the dataflow between a control (Numeric) and an indicator (Slide).

In the first five weeks of ECE200, students are introduced to the following LabVIEW components and features:

1. Standard and custom controls and indicators
2. Library functions provided by LabVIEW
3. Macros or functions (sub-programs)
4. While and for loops
5. Shift registers
6. Case and decision structures
7. Sequences
8. Arrays and clusters
9. Charts and graphs
10. Character strings
11. File input and output

Through these eleven lessons, the students are assigned thirteen homework assignments. Afterwards, they must take an hour exam on LabVIEW and complete a short mini-project (such as a English-Pirate translator) using LabVIEW.

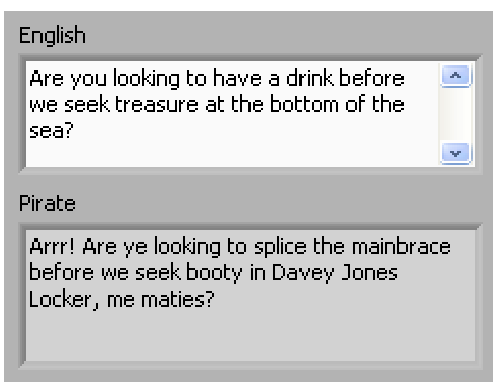

Figure 6. Example of a short LabVIEW project undertaken by ECE200 students prior to developing engineering games.

5. COMPUTER GAME REQUIREMENTS

For their final LabVIEW project (the K-5 engineering game), the ECE200 undergraduate game developers were given the following four requirements.
First, the game must incorporate a minimum number of LabVIEW features including:

1. Binary, numeric, and string controls and indicators
2. Functions from LabVIEW's library
3. At least one while or for loop
4. At least one shift register
5. Arrays and clusters

These requirements would serve to demonstrate the ECE undergraduate's mastery of the essential elements in any LabVIEW program.
Second, the game must feature an application-based lesson related to the engineering course work in GE100 / ECE110. Students were asked to prioritize three preferred engineering lessons for their game. This was done so that the professor could ensure that a variety of different engineering lessons would be supported by the new games. The engineering lessons included:

1. Ohm's law
2. Digital logic design and analysis
3. Analog circuit design and analysis
4. Power transmission
5. Transistor applications

This second requirement reinforces earlier coursework and builds a stronger foundation for the undergraduates as they progress through their engineering curriculum.

Third, the programs must meet certain specifications. The games must be instructional and informational. The games must teach players about the selected engineering topic. In order to score well and "win" the game, players must demonstrate proficiency in the lesson. The games must take ten to fifteen minutes to play and be targeted to 4th and 5th graders. Anecdotally, the games must be "fun."

Finally, the games would be developed with the following schedule:

Date	Event
25-Sep	Projects selected
12-Oct	Front panel (user interface) due
30-Oct	Revision 1.0 of program due
13-Nov	Peer reviews due
30-Nov	Revision 2.0 of program due
02-Dec	Peer reviews due
07-Dec	Final game due

6. RESULTS

A total of thirty-one games have been developed by the ECE200 students. Overall, the quality and playability of the games was high, but not all of the games were usable (see Figure 7). Twenty-three of the games were of a quality suitable for use in our College of Engineering's primary student outreach programs. Two of the games were incomplete and did not function or run. Of the remaining projects, three were deemed too complex for our elementary student audience, and three lacked sufficient documentation for the young students to work independently.

Games that did not function
Two of the thirty-one games did not function, but they failed for similar reasons. Details of each of the failed games are below.

The scope of the first failed game was very aggressive (see Figure 8). As a result, the students were not able to complete the game according to the assigned schedule (see Figure 9). In addition, if the game had been completed, it would probably have been deemed too complex for the elementary student outreach program.

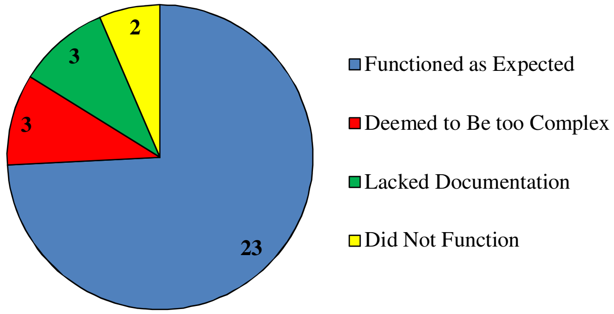

Figure 7. High-level results of thirty-one LabVIEW games developed by undergraduate engineering students to introduce primary students to engineering.

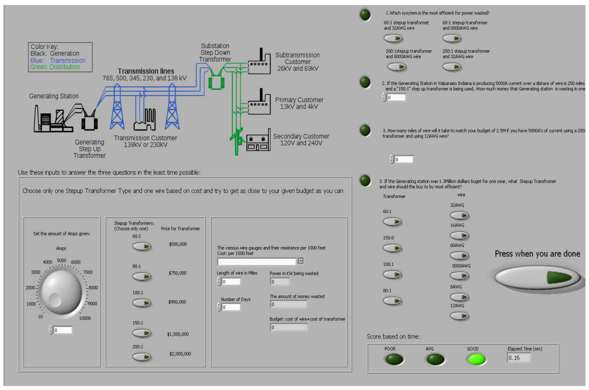

Figure 8. One of the failed games had a very aggressive scope.

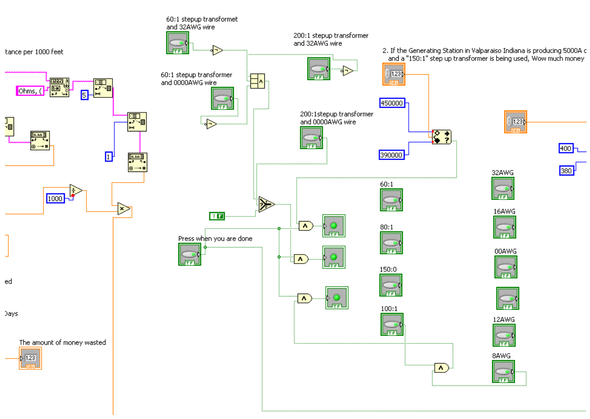

Figure 9. The block diagram of the failed game in Figure 8 illustrates that the undergraduate students did not complete the assignment.

The second failed game had a very simple scope. Given two variables in Ohm's Law (*Voltage* = *Current* times *Resistance*, *V=IR*), solve for the third. However, the undergraduate student programmers for this game did not take the assignment seriously and failed to complete it on time.

Games that lacked documentation

Three of the thirty-one games lacked sufficient on-screen documentation to be usable in our College of Engineering's elementary student outreach program. It is imperative that the games be very intuitive for the young students, or two problems may occur. First, it may take too long to explain how the games work to the elementary students, taking valuable time away from the outreach program. Second, the young students could be overwhelmed or frustrated by the lack of documentation and lose interest in the learning opportunity.

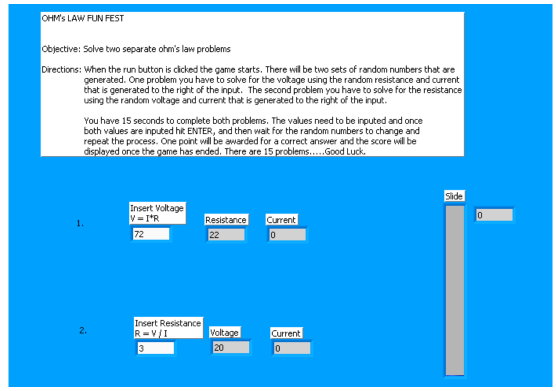

Figure 10. The second failed game had a relatively small scope and was not completed by the assigned date.

The first game required the user to design a multi-level Boolean logic circuit to "crack" a safe (see Figure 11). The second implemented a version of Tic-Tac-Toe (or Noughts and Crosses) where users claimed squares on the board by solving Boolean equations (Figure 12). Finally, the third required users to apply Ohm's Law to navigate a series of stop lights (Figure 13). In all three cases, the games and their documentation were viewed as incomprehensible by our elementary student testers. However, both games were playable by the elementary students with additional (five to ten minutes) verbal instruction.

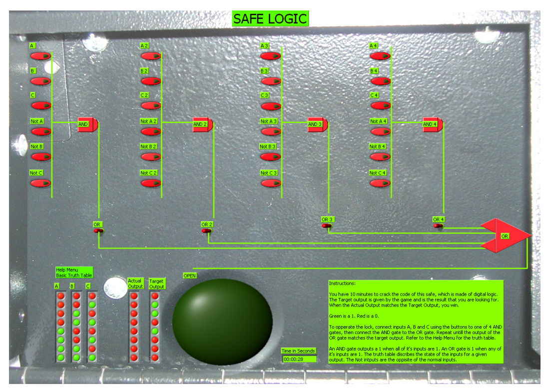

Figure 11. One of the games that lacked sufficient on-screen documentation.

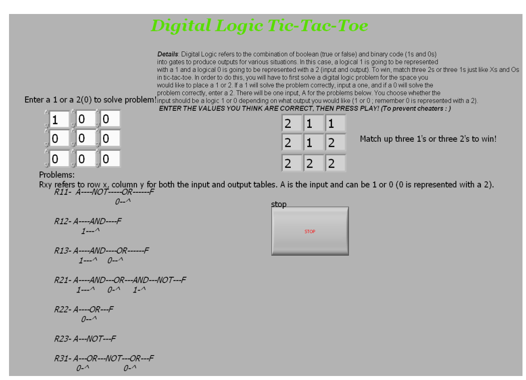

Figure 12. One of the games that lacked sufficient on-screen documentation.

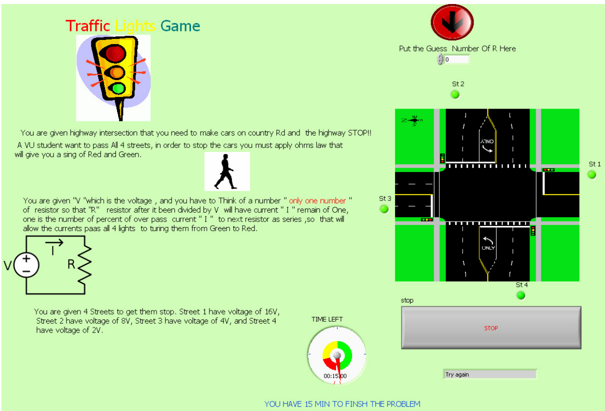

Figure 13. One of the games that lacked sufficient on-screen documentation.

Games deemed to be too complex

Three of the games developed by the undergraduate student programmers were simply too complex for elementary students to play in the ten to fifteen minute time target. The reason for this was two-fold. First, each game contained a relatively advanced topic from the freshmen engineering classes (transistor biasing, equivalent resistances, and power transmission). Second, each of these topics was not a significant topic in our College of Engineering's elementary student outreach program. Therefore, these topics will not be included in future game developments in the near future.

Games that functioned as expected

A good majority (74%) of the games met the expectations set forth by the instructor. As a prerequisite for this classification, the games had to meet the requirements outline in Section 5:

1. The game had to incorporate a minimum number of LabVIEW features.
2. The game had to feature an application-based lesson related to the undergraduates' prior engineering classes.
3. The game had to be instructional and feature a way to score or win by demonstrating proficiency in the subject matter within 10-15 minutes.
4. The games had to meet the assigned schedule.

While it is not practical to include all twenty-three, details and screen shots of a few of the games are shown below. (Several of the undergraduate students used characters and images from outside sources in their games. The importance of "fair use" and respect for copyrighted material needs to be continually impressed upon our undergraduate students.)

The first game introduced elementary students to Ohm's Law (Figure 14). Users were challenged to answer a series of three question challenges within twenty seconds. The faster the students answered correctly, the more points they scored. When the students had scored a sufficient number of points, the game was won. Test groups of elementary students rated this game highly because it was based upon a character story, and it provided feedback on correct intermediate results as the game progressed.

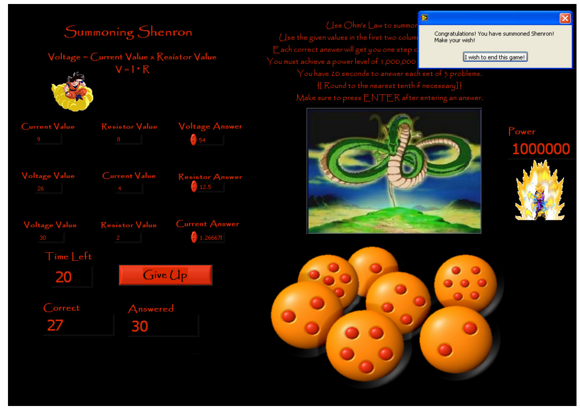

Figure 14. Game introducing elementary students to Ohm's Law.

The next game introduced elementary students to Kirchhoff's Current Law (Figure 15). Users were challenged solve a series of increasingly difficult puzzles based upon the concept of current flow: Any current flowing into a node must also flow out of the node. As the user solves each puzzle, the user gets one step closer to saving Kirchhoff. Test groups of elementary students rated this game highly because it was based upon a character story, and it again provided feedback on correct intermediate results as the game progressed.

Figure 15. Game introducing elementary students to Kirchhoff's Current Law.

The third game also introduced elementary students to Ohm's Law (Figure 16). The game's theme was "Who Wants to Be a Millionaire?" Users answered increasingly difficult questions about Ohm's Law. Test groups of students rated this game highly because it was again based upon a character story and provided feedback on correct intermediate results as the game progressed. Authentic music and audio files added to the realism of the game.

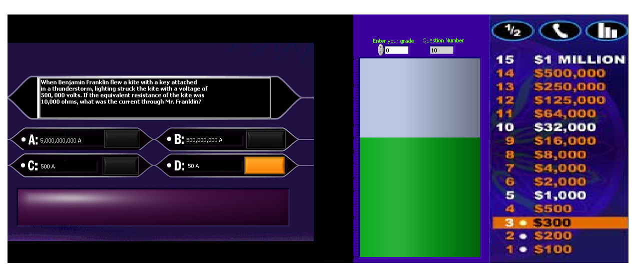

Figure 16. Game introducing elementary students to Ohm's Law.

The fourth game also introduced elementary students to Ohm's Law (Figure 17). The game's pirate theme was a hit with the student test groups. Users were given two of three variables in Ohm's Law and asked to solve for the third. Test groups, however, did not like the way the game ended. Based upon the build-up of the game, they expected a more spectacular finish. Also, the game required answers to be given to two decimal points (for example, 0.78), and calculators were not always available.

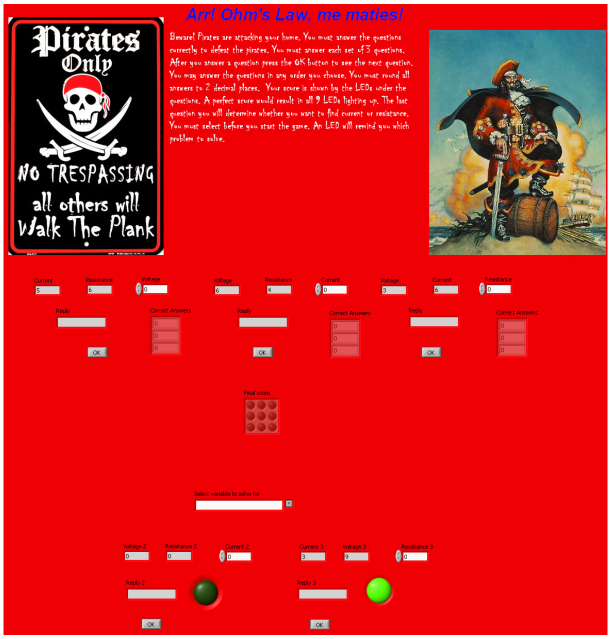

Figure 17. Game introducing elementary students to Ohm's Law.

The last game introduced elementary students to binary numbers (Figure 18). Users were given a binary number and asked to solve for its decimal equivalent. Test groups liked the fact that it tested their math abilities and that the game progressed through three increasingly difficult rounds. However, the lack of a game theme caused the elementary test groups to lose interest in the game quickly. The lack of a calculator for the final, most difficult round was also seen as a shortcoming by several of the students.

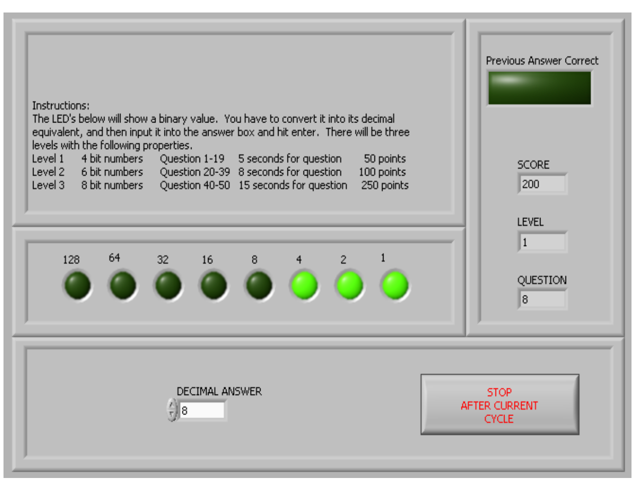

Figure 18. Game introducing elementary students to binary numbers.

In general the feedback from the elementary school student test groups was:

1. Most of the games had timed activities, and the time was often inadequate. Future games should include a master time variable that can be easily accessed to slow down or speed up the game.
2. Too many calculations required a calculator. Many engineering undergraduates can perform two or three digit operations quickly in their head or on pencil and paper, but elementary students cannot. The games that featured a built in calculator were generally viewed as better.
3. As a whole, the elementary students expressed a desire for more action in the games. The test groups realized that these were "thinking" games and not "shooting" or "driving" games like they have often played at home, but they still wanted to see more motion on the computer screens. One student suggested a short video as an introduction or reward would be sufficient. The rest of the test group agreed.
4. The games that provided feedback on correct intermediate answers were generally viewed much more positively.
5. The test groups thought the games were more fun when they were based upon characters or themes that were familiar to them. In the future, ECE200 will introduce additional instruction on fair use and copyrighted materials, while still seeking to provide the themes and stories that interest the children.

The LabVIEW game developers in ECE200 also provided very enthusiastic support for the parallel learning opportunity. The undergraduate students were asked four questions related to the game development. They were asked to compare their experience developing the LabVIEW game to developing a data acquisition and control project, which had been done in previous years. The questions were answered using a Likert scale (5 means strongly agree, 3 means neutral, 1 means strongly disagree). The questions were:

1. I had more fun working with the game than I would have had with a data acquisition and control program.
2. It was more interesting working on the game than I would have had on a data acquisition and control program.
3. Because I found it more fun and more interesting, I spent more time developing the game than I would have spent on a data acquisition and control program.
4. Because I spent extra time on developing the game, I learned a lot more about LabVIEW than I would have with a data acquisition and control program.

Figure 19 shows a histogram of the responses to the four questions. Table 1 shows the average scores for the four questions. While the number of responses was small (thirty-three), we did calculate the standard deviation. As Figure 19 shows, there were no responses of 1 (Strongly Disagree) or 2 (Disagree), and limited number of responses of 3 (Neutral). Therefore, the standard deviation essentially encompasses all of the responses based upon averages of 4.39 - 4.61. Based upon this student feedback, it is very safe to state that the use of a game development as an alternative to the data acquisition and control program was strongly preferred by the students and gave them the impression that they had learned more.

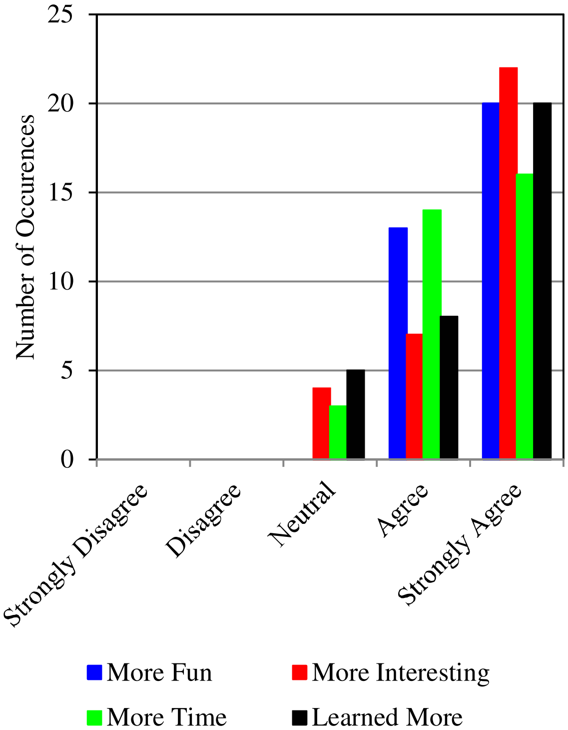

Figure 19. Histogram of student feedback showing strong favorable response to the use of game development as a learning technique for LabVIEW.

Table 1
Students' Perception of Game Development
Compared to Data Acquisition and Control Program

	Average	Std. Dev.
I had more fun with the game	4.61	0.49
The game was more interesting	4.54	0.70
I spent more time on the game	4.39	0.65
I learned more developing the game	4.48	0.70

However, student perceptions can be misleading. Therefore, the students were given a standardized LabVIEW Fundamentals Exam developed and administered by National Instruments. The average on the standardized exam by students developing a LabVIEW based game was 86%. This is 11% higher than the average by the students developing a data acquisition and control program (75%), rather than the LabVIEW game, as a LabVIEW project.

7. CONCLUSIONS

Our College of Engineering is working to revitalize the development of pre-engineering resources that are appropriate for elementary school students. We have developed twenty-three short computer games intended to introduce primary students to engineering. The game development was accomplished in parallel to reinforcing undergraduate student knowledge of engineering fundamentals and computer programming. The games have proven to be a fun and rewarding opportunity for both the elementary students and the undergraduate students to better understand the role and work of engineers in the 21st century.

8. REFERENCES

[1] J. L. DeGrazia, J. F. Sullivan, L. E. Carlson, D. W. Carlson, "A K-12/University Partnership: Creating Tomorrow's Engineers," *Journal of Engineering Education*, October, 2001, pp. 557 - 563.

[2] C. P. Lachapele and C. M. Cunningham, "Engineering Is Elementary: Children's Changing Understandings of Science and Engineering," Proceedings of the American Society for Engineering Education Annual Conference, Honolulu, HI, June 24-27, 2007.

[3] C. B. Rogers, K. Wendell, J. Foster, "A Review of the NAE Report, Engineering in K-12 Education," *Journal of Engineering Educatio*n, April, 2010, pp. 179 - 181.

[4] P. Cantrell, J. Ewing-Taylor, "Exploring STEM Career Options through Collaborative High School Seminars," *Journal of Engineering Education*, July, 2009, pp. 295 - 303.

[5] M. J. Nathan, N. A. Tran, A. K. Atwood, A. Prevost, L. A. Phelps, "Beliefs and Expectations about Engineering Preparation Exhibited by High School STEM Teachers," *Journal of Engineering Education*, October 2010, pp. 409 - 426.

[6] M. Reisslein, R. Moreno, G. Ozogul, "Pre-college Electrical Engineering Instruction: The Impact of Abstract vs. Contextualized Representation and Practice on Learning," *Journal of Engineering Education*, July 2009, pp. 225-235.

[7] G. M. Nicholls, H. Wolfe, M. Besterfield-Sacre, L. J. Shuman, "Predicting STEM Degree Outcomes Based on Eighth Grade Data and Standard Test Scores," *Journal of Engineering Education*, July, 2010, pp 209 - 223.

[8] N. A. Tran, M. J. Nathan, "Pre-College Engineering Studies: An Investigation of the Relationship Between Pre-college Engineering Studies and Student Achievement in Science and Mathematics," *Journal of Engineering Education*, April 2010, pp. 143 - 157.

[9] D. M. Beck, G. Vrabel, M. M. Budnik, "Program for Gifted and Talented Elementary School Children, Proceedings of the Frontiers in Education Conference, San Antonio, TX, October 18-21, 2009.

[10] M. M. Budnik, H. A. Budnik, "Engineering Computer Games: A Parallel Learning Opportunity for Under-graduate Engineering and Primary (K-5) Students," Proceedings of the International Conference on Education, and Information Systems, Technologies, and Applications (EISTA), June 29 - July 2nd, 2010, Orlando, FL, pp. 284 - 289.

[11] Valparaiso University, **General Catalog 2010-2011**, Valparaiso University Press. 2010.

[12] R. H. Bishop, **LabVIEW 2009 Student Edition**, Prentice Hall: Upper Saddle River, NJ, 2010.