2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #40253

How Maker Culture Improves Students' Learning Experiences in Computing Programs

Dr. Yonghui Wang, Prairie View A&M University

Dr. Yonghui Wang received his B.S. in Optoelectronics from Xidian University in 1993, his M.S. in electrical engineering from Beijing Polytechnic University in 1999, and his Ph.D. in computer engineering from Mississippi State University in 2003. He is currently with the Department of Computer Science at Prairie View A&M University, Prairie View, TX.

Dr. Suxia Cui, Prairie View A&M University

Suxia Cui is an associate professor in the Department of Electrical and Computer Engineering at Prairie View A&M University (PVAMU). She joined PVAMU right after she obtained her Ph.D. degree in Computer Engineering from Mississippi State University in 2003. Her research interests include image and video processing, data compression, wavelets, computer vision, remote sensing, and computing education. Her projects are currently funded by NSF, United States Department of Agriculture, and Department of Education.

Dr. Bugrahan Yalvac, Texas A&M University

Bugrahan Yalvac is an associate professor of science and engineering education in the Department of Teaching, Learning, and Culture at Texas A&M University, College Station. He received his Ph.D. in science education at the Pennsylvania State University i

Dr. Wei Zhan, Texas A&M University

Dr. Wei Zhan is a Professor of Electronic Systems Engineering Technology at Texas A&M University. Dr. Zhan earned his D.Sc. in Systems Science from Washington University in St. Louis in 1991. From 1991 to 1995, he worked at University of California, San

How Maker Culture Improves Students' Learning Experiences in Computing Programs

Abstract

Although undergraduate computing coursework covers the curriculum and most basic and advanced concepts, there still exists a big gap between what is learned in the class and its application in solving different real-life problems. Additionally, traditional teacher-centered instruction may not attract students well, and may further lead to low retention rates. Innovative pedagogies are highly demanded to enhance computing-related teaching and learning efficiencies. Based on these observations, the objectives of this study are to create a project-based maker concept instruction that will help bridge the gap between classroom studies and real-life problem-solving techniques and to cultivate a maker culture that will help form students' learning-by-doing habits and enhance their life-long learning skills. This will help the students understand and apply their computer science knowledge in solving real-life problems and prepare them better for their future technical careers.

Learning-by-doing / maker concepts are introduced and delivered in lower-level computing courses. Student-chosen real-application-based project assignments are designed and assigned for students to solve real-life problems. Courses like Computer Science I and Computer Organization are chosen to implement this innovative instruction. Students choose their real-application-based projects and work in teams. Their projects must relate highly to the knowledge introduced in the class. For example, in Computer Organization class, students are required to design a simple computer. Students need to design their computer's instruction set architecture, its data path, and its final implementation. In the process, students actively practice their lifelong learning skills by studying more rigorously on details about computer architecture and organization. This learning-by-doing process significantly enhances students' understanding of the concepts learned in class. It dramatically helps students in understanding the course concepts. To further stimulate students' learning interests, a mini maker faire is organized at the end of the semester for each course to show off students' projects. Students communicate with and learn from their peers in a maker faire environment. Such a learning environment makes them more confident about what they learned and how they can benefit society. Students' survey shows that this innovative pedagogy greatly helps our students and meets our expectations.

Introduction

In the last decade, with the advancement of artificial intelligent technology, more and more previously unthinkable applications and services become possible. To meet this trend, more and more new technical positions are created and are ready to be filled. Skilled and well-prepared engineers are highly demanded by such newly emerging positions. Computing programs in US universities cannot produce enough qualified graduates to fill these positions. To make the problem even worse, computer programs suffer high dropout and failure rates, mainly due to the reason that students are unprepared and lose their interest in their entry-level courses [1, 2, 3, 4, 5]. In fact, a significant shortage of skilled computer science graduates is observed and will remain for the next decade [6, 7, 8]. The US Labor Department estimates that the global shortage of software engineers may reach 85.2 million by 2030 [6]. Job openings for software developers, quality assurance analysts, and testers are projected to grow by 22% year over year between 2020 to 2030 [6].

To alleviate this problem, a group of researchers works together to introduce innovative teaching approaches in entry-level computing courses. Learning-by-doing or maker movement concepts are introduced to the classes to attract students to stay with their computing programs. Students are required to work on their self-selected projects and then show off their projects in a maker faire like environment.

Maker Movement

As explained at techopedia.com, the maker movement is "primarily the name given to the increasing number of people employing do-it-yourself (DIY) and do-it-with-others (DIWO) techniques and processes to develop unique technology products. Generally, DIY and DIWO enable individuals to create sophisticated devices and gadgets, such as printers, robotics, and electronic devices, using diagrammed, textual, and or video demonstrations. With all the resources now available over the Internet, virtually anyone can create simple devices, which in some cases are widely adopted by users." The maker culture emphasizes informal, networked, peer-led, and shared learning experiences motivated by fun and self-fulfillment [9]. The maker culture emphasizes hands-on skills and learning-through-doing in a social environment. These skills and learning philosophies especially fit the teaching/learning objectives for engineering and computer science students. With students becoming increasingly disengaged from STEM subjects in formal educational settings, the introduction of a maker culture to the classroom has the potential to create new pathways into topics that will make computing more interesting and attractive to learners. Based on these observations, the authors research and explore effective methods to cultivate a maker culture in our computing programs, and support a mini maker faire for each participated course.

Computing Programs

There are two computing programs in Prairie View A&M University: Computer Engineering and Computer Science. Both programs are committed to providing the highest quality instruction to the students; conducting leading-edge research in computer science can engineering; and providing leadership and service to our professional communities. Three entry-level courses from the two computing programs are selected to participate in this project. Two undergraduate courses from Computer Science (CS) Department are chosen.

- COMP 1336 Computer Science I: 3 semester hours. Introduction to modern problem-solving and programming methods. Special emphasis is placed on using critical thinking, effective communication, and empirical and quantitative skills to design and implement robust and easily maintainable programs in a high-level, object-oriented language such as C++ to include external files, control structures, loops, scope, functions, output formatting, inline functions and function templates, enumerated data types, arrays, structures, exception handling.
- COMP 2319 Computer Organization: 3 semester hours. This course examines computer organization and architecture, based on the idea that a computer can be regarded as a hierarchy of levels. Topics include gates, logic circuits, processors, memory, I/O, micro-architecture, instruction set architecture, operating system machine level, and assembly language.

One computer engineering undergraduate course participates in this project.

- ELEG 1304 Computer Applications in Engineering: 3 semester hours. Fundamentals of C++ Programming language and MATLAB applications software. The logic of algorithms, flowcharts, program looping, conditional statements, arrays, functions and pointers, Engineering applications and team projects.

Approaches

To reach the goal of this research, in each participating course, two activities are implemented, the project design and the mini-maker faire. At the beginning of the class, students are instructed to find course-related projects, that interest them and can solve real application problems. Students form their teams and work within a team environment. Instructor approval must be obtained before starting their projects. Maker concepts must be considered during their project design processes. Maker space and devices, such as a 3D printer and a milling machine, are provided and students can access the facilities with the supervision of faculty and technicians. At the end of the semester, for each course, a mini maker faire is organized to let the project teams present their projects. Different from a traditional classroom project presentation, in a mini maker faire setting, students can communicate with and learn from their peers in a friendly and informal environment. Such a learning environment makes them more confident about what they learned and how they can benefit society.

Case Study

For all participating courses, the final projects are enhanced with the concepts of the maker movement. Each project is designed to have three deliverables: 1) Design the final project by integrating the scheme of maker movement; 2) Present the project in a mini maker fair at the end of the semester; and 3) Summarize the learning from the maker movement as one section in final project report after reading necessary concepts and examples.

Computer Science Courses

The participating computer science courses are COMP 1336 Computer Science I and COMP 2319 Computer Organization. These are required courses for all computer science majors. They are low-level courses and the prerequisites for most high-level CS courses. Student projects are major practices for both courses. These courses perfectly fit the objective of this research to attract and prepare the students to stay with the CS program. In one semester, there were 32 students enrolled in COMP 1336 and 32 students enrolled in COMP 2319. To promote teamwork, groups were formed, each with 3 or 4 students. Some example COMP 1336 projects are listed in Table 1.

Table 1: Example COMP 1336 Computer Science I Final Projects

No.	Topic	Description
1	Spanish Helper	Program for Helping Users to Learn and Speak Spanish
2	Sentence Summary	Word count, alphabets count and replace words
3	Student Grading	Accept Student names, courses, and scores. Print names and grades
4	Multiplication table	Generate a well-formatted multiplication table

The final project for COMP 2319 is to design and implement a simple computer, which can run simple programs. It is divided into 3 parts. In part 1, students are required to design and encode the instruction set for the computer. Students need to determine which registers they want to use, create a sequence of machine instructions, and produce a unique binary encoding for each instruction. Students need to consider the length of instruction encoding and the size of addressable memories. In the 2nd part, students are required to design a datapath and control instructions that will execute their instruction set, which is designed in part 1. By the time this project is assigned, the course has covered block diagrams of datapaths and microassembly control code. The third subproject requires students to program a simulator for their computers, using their control code from the second subproject. To demonstrate that their simulators work correctly, the project requires students to execute a program segment on their simulator with varying data sizes.

For both courses, mini maker faire was organized at the end of the semester, as shown in Figure 1.

Figure 1: Computer Science team project mini maker faire.

Computer Engineering Course

The participating computer engineering course is ELEG 1304 Computer Applications in Engineering. This is a required course for the Computer Engineering degree. It is a freshmen-level course. It has an emphasis on student projects, which are feasible for “maker movement” implementation. Thus, final projects are enhanced with the contents of the maker movement in this course. This course is also a first-semester C++ course for all engineering students. In one semester, there were 17 students enrolled. To promote teamwork, six groups were formed, each with 2 or 3 students. The project topics are listed in Table 2.

Table 2: ELEG 1304 Computer Applications in Engineering Final Project List

No.	Topic	Description
1	Scrabble game	Multiple players can play scrabble game together
2	Guessing game 1	Guess a pre-defined number with limited trial
3	Space bomber	Fire a cannon ball to hit a target
4	Tic tac toe	Player will play with computer for the popular tic tac toe game
5	Guessing game 2	Guess a famous film character based on limited hints
6	Board game	Calculate distance between two random inputted points

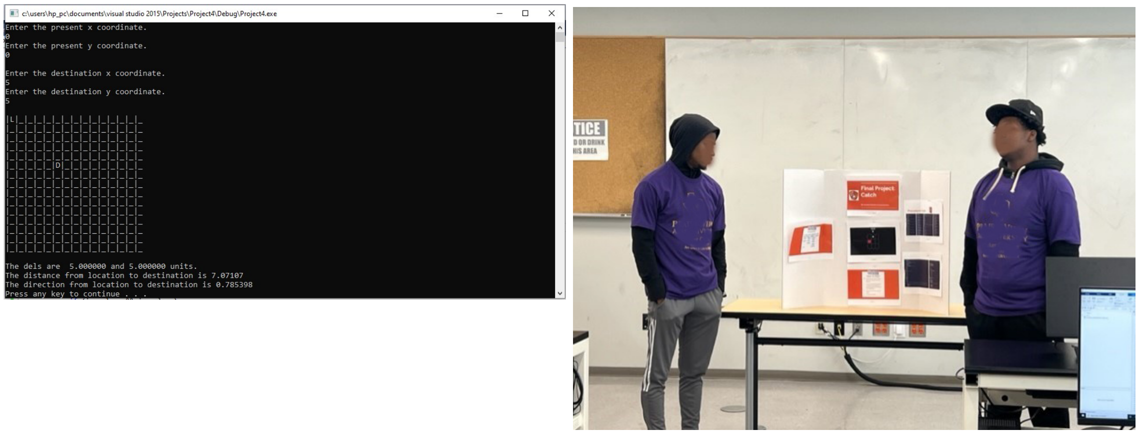
One example project is shown below:

Board game: This is an example of building a graphic game board for C++. The characters used as markers on the board can be changed and the board can be adapted to different types of logic, size, and number of symbols on the board. Figure 2(a) is an example of using the board to show travel from two locations. At the end of the semester, there was a mini maker faire for all the groups to present their work using a tri-fold poster, as shown in Figure 2(b). And the competition includes playing each other’s games. Credits were given to the best design and best play. The top three teams obtained awards.

Results

To evaluate the effectiveness of the approaches, student surveys were administered and students’ feedback was collected for each participating class. Tables 3-6 tabulate the survey results of COMP 1336 Computer Science I. Thirty-two students completed the survey. The failure rate comparison is shown in Table 7. ‘D’, ‘F’, and ‘W’ (withdrew) are considered failure grades. Tables 8-11 tabulate survey results for COMP 2319 Computer Organization. Thirty-two students completed the survey. Survey results of ELEG 1304 Computer Applications in Engineering are listed in Tables 12-15. Eleven students completed the survey.

From the survey results, we can see that both the awareness about and the interest in learning-by-doing and maker movement increase significantly after students completed the courses. From Tables 5, 10, and 14, the results show that the maker movement projects and mini maker faire can dramatically help students to gain a high-level understanding of different topics.



(a) (b)

Figure 2: (a) Screenshot for board game implementation and (b) A sample team project presentation.

Tables 6, 11, and 15 show that most students agreed or strongly agreed that the learning-by-doing / maker movement helped them in learning course concepts and they would like to see such concept be introduced in more classes. Table 7 shows that the proposed approaches significantly reduce students' failure rate for COMP 1336 from over 30% to 11.8%.

Table 3: COMP 1336 survey results for before and after comparison for awareness about learning-by-doing

	Never heard anything about	Only heard the term	Know only a few things about	Know some basic concepts	Know well
Before	22 (68.8%)	3 (9.4%)	3 (9.4%)	4 (12.5%)	0
After	4 (12.5%)	8 (25%)	5 (15.6%)	9 (28.1%)	6 (18.8%)

Table 4: COMP 1336 survey results for before and after comparison for interest in learning-by-doing

	Never heard about	Not interested in	A little interested in	Interested in	Very interested in
Before	15 (46.9%)	2 (6.3%)	4 (12.5%)	10 (31.3%)	1 (3.1%)
After	2 (6.3%)	1 (3.1%)	12 (37.5%)	11 (34.4%)	6 (18.8%)

Conclusion

In this paper, a student-centered approach is designed and delivered in lower-level computing courses. Student-chosen real-application-based project assignments are assigned and designed for students to solve real-life problems. Courses like Computer Science I and Computer Organization

Table 5: COMP 1336 survey results for gaining an understanding of different topics

Topic	A Great Deal	A Lot	Somewhat	A Little	Not at all
C++ Syntax and Semantics	7 (21.9%)	16 (50%)	8 (25%)	1 (3.1%)	0
C++ Program Structures	9 (28.1%)	14 (43.8%)	9 (28.1%)	0	0
C++ Functions	12 (37.5%)	14 (43.8%)	6 (18.8%)	0	0

Table 6: COMP 1336 survey results for the level of agreement with the statements

Statement	Strongly Agree	Agree	Average	Disagree	Strongly Disagree
1	10 (31.3%)	18 (56.3%)	4 (12.5%)	0	0
2	11 (34.4%)	16 (50%)	5 (15.6%)	0	0
3	13 (40.6%)	11 (34.4%)	6 (18.8%)	1 (3.1%)	1 (3.1%)

Statement 1 Learning-by-Doing / Maker Movement helped me in learning the concepts in this class more effectively.

Statement 2 I would like Learning-by-Doing concept to be introduced in more classes.

Statement 3 Teamwork helped me in learning more effectively.

Table 7: COMP 1336 failure rate comparison for COMP 1336 Computer Science I.

Previous Semester	Same Semester Without Maker	Course with Maker
38.2%	34.5%	11.8%

Table 8: COMP 2319 survey results for before and after comparison for awareness about learning-by-doing

	Never heard anything about	Only heard the term	Know only a few things about	Know some basic concepts	Know well
Before	19 (59.4%)	5 (15.6%)	6 (18.8%)	0	2 (6.3%)
After	1 (3.1%)	2 (6.3%)	3 (9.4%)	18 (56.3%)	8 (25%)

Table 9: COMP 2319 survey results for before and after comparison for interest in learning-by-doing

	Never heard about	Not interested in	A little interested in	Interested in	Very interested in
Before	11 (34.4%)	3 (9.4%)	11 (34.4%)	4 (12.5%)	3 (9.4%)
After	0	2 (6.3%)	10 (31.3%)	12 (37.5%)	8 (25%)

Table 10: COMP 2319 survey results for gaining an understanding of different topics

Topic	A Great Deal	A Lot	Somewhat	A Little	Not at all
Instruction Set Design	3 (9.4%)	14 (43.8%)	11 (34.4%)	4 (12.5%)	0
Data Path Design	3 (9.4%)	17 (53.1%)	9 (28.1%)	1 (3.1%)	2 (6.3%)
How Computer Works	9 (28.1%)	13 (40.6%)	9 (28.1%)	1 (3.1%)	0

Table 11: COMP 2319 survey results for the level of agreement with the statements

Statement	Strongly Agree	Agree	Average	Disagree	Strongly Disagree
1	8 (25%)	19 (59.4%)	4 (12.5%)	0	1 (3.1%)
2	8 (25%)	15 (46.9%)	7 (21.9%)	0	2 (6.3%)
3	11 (34.4%)	13 (40.6%)	7 (21.9%)	0	1 (3.1%)

Statement 1 Learning-by-Doing / Maker Movement helped me in learning the concepts in this class more effectively.

Statement 2 I would like Learning-by-Doing concept to be introduced in more classes.

Statement 3 Teamwork helped me in learning more effectively.

Table 12: ELEG 1304 survey results for before and after comparison for awareness about learning-by-doing

	Never heard anything about	Only heard the term	Know only a few things about	Know some basic concepts	Know well
Before	6 (54.5%)	5 (45.5%)	0	0	0
After	1 (9%)	1 (9%)	1 (9%)	5 (545.5%)	3 (27.3%)

Table 13: ELEG 1304 survey results for before and after comparison for interest in learning-by-doing

	Never heard about	Not interested in	A little interested in	Interested in	Very interested in
Before	3 (27.3%)	2 (18.2%)	4 (36.4%)	2 (18.2%)	0
After	1 (9%)	0	2 (18.2%)	5 (45.5%)	3 (27.3%)

are chosen to implement this innovative instruction. Students choose their real-application-based projects and work in teams. Their projects must highly relate to the knowledge introduced in the class. In the process of completing their projects, students need to practice their life-long learning skills by studying more details about computing concepts. This learning-by-doing process significantly enhances students' understanding of the concepts learned in the class. To further stimulate students' learning interests, a mini-maker faire is organized at the end of the semester

Table 14: ELEG 1304 survey results for gaining an understanding of different topics

Topic	A Great Deal	A Lot	Somewhat	A Little	Not at all
Instruction Set Design	2 (18.2%)	5 (45.5%)	3 (27.3%)	1 (9%)	0
Data Path Design	2 (18.2%)	4 (36.4%)	4 (36.4%)	1 (9%)	0
How Computer Works	4 (36.4%)	3 (27.3%)	4 (36.4%)	0	0

Table 15: ELEG 1304 survey results for the level of agreement with the statements

Statement	Strongly Agree	Agree	Average	Disagree	Strongly Disagree
1	3 (27.3%)	8 (72.7%)	0	0	0
2	3 (27.3%)	7 (63.6%)	1 (9%)	0	0
3	4 (36.4%)	4 (36.4%)	2 (18.2%)	0	0

Statement 1 Learning-by-Doing / Maker Movement helped me in learning the concepts in this class more effectively.

Statement 2 I would like Learning-by-Doing concept to be introduced in more classes.

Statement 3 Teamwork helped me in learning more effectively.

for each course to show off students' projects. Students communicate with and learn from their peers in a mini-maker faire environment. Such a learning environment makes them more confident about what they learned and how they can benefit society. Students' survey shows that this innovative pedagogy greatly helps our students and meets our expectations.

References

[1] "Why Do Students Drop Out of Tech Studies?" https://wawiwa-tech.com/blog/why-do-students-drop-out-of-tech-studies/#:~:text=Computer\%20Science\%20Tops\%20the\%20Dropout,highest\%20dropout\%20rate\%20\%E2\%80\%93\%209.8\%25., 2020, [Online; accessed 26-February-2023].

[2] K. Flinders, "Computer science undergraduates most likely to drop out," https://www.computerweekly.com/news/252467745/Computer-science-undergraduates-most-likely-to-drop-out, 2019, [Online; accessed 26-February-2023].

[3] L. E. Margulieux, B. B. Morrison, and A. Decker, "Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples," *International Journal of STEM Education*, vol. 7, no. 19, 2020. [Online]. Available: https://doi.org/10.1186/s40594-020-00222-7

[4] "Undergraduate Retention and Graduation Rates," https://nces.ed.gov/programs/coe/indicator/ctr/undergrad-retention-graduation, May 2022, [Online; accessed 26-February-2023].

[5] M. Haungs, C. Clark, J. Clements, and D. Janzen, "Improving first-year success and retention through interest-based cs0 courses," in *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 589–594. [Online]. Available: https://doi.org/10.1145/2157136.2157307

[6] T. Phillips, "Is There a Shortage of Developers? Developer Shortage Statistics in 2022," https://codesubmit.io/blog/shortage-of-developers/, 2022, [Online; accessed 26-February-2023].

[7] P. Kenkare, "Faced with a desperate skills shortage, tech leaders plead for computer science reform in schools," https://www.zdnet.com/education/computers-tech/tech-leaders-call-for-more-computer-science-opportunities-in-schools/, 2022, [Online; accessed 26-February-2023].

[8] R. Huxford, "Software Developers and Engineers Shortage in the US: The case for Global Talent," https://www.revelo.com/blog/software-developer-shortage-us, 2022, [Online; accessed 26-February-2023].

[9] E. FitzGerald, "Innovating Pedagogy 2013: Open University Innovation Report 2," https://www.academia.edu/4453693/Innovating_Pedagogy_2013_Open_University_Innovation_Report_2, 2013, [Online; accessed 26-February-2023].