2017

The Application of Second Language Acquisition to Programming Language Study

Lulu Sun
Embry-Riddle Aeronautical University, sunl@erau.edu

Christina Frederick
Embry-Riddle Aeronautical University

Li Ding
Embry-Riddle Aeronautical University, dingl@erau.edu

Rebecca Rohmeyer
Embry-Riddle Aeronautical University, rohmeyer@my.erau.edu

The Application of Second Language Acquisition to Programming Language Study

Dr. Lulu Sun, Embry-Riddle Aeronautical Univ., Daytona Beach

Lulu Sun is an associate professor in the Engineering Fundamentals Department at Embry-Riddle Aeronautical University, where she has taught since 2006. She received her Ph.D. degree in Mechanical Engineering from University of California, Riverside, in 2006. Before joining Embry-riddle, she worked in the consulting firm of Arup at Los Angeles office as a fire engineer. Her research interests include second language acquisition in programming languages, and online course design. She is a member of the American Society for Engineering Education.

Dr. Christina Frederick, Embry-Riddle Aeronautical Univ., Daytona Beach

Dr. Frederick is currently a Professor and Graduate Program Coordinator in the Human Factors and Systems Department at Embry-Riddle Aeronautical University in Daytona Beach, Florida. Dr. Frederick received her Ph.D. in 1991 from the University of Rochester with a major in Psychological Development. She previously taught at the University of Rochester, Southern Utah University and the University of Central Florida. In 2000, Dr. Frederick joined the Human Factors and Systems Department at Embry-Riddle, where her work focused on applied motivation and human factors issues in aviation/aerospace. Dr. Frederick also served in various roles in University administration between 2004-2012, including Vice President for Academics and Research. Dr. Frederick's current research interests examine how individual differences interact with technology to enhance educational engagement and performance. Dr. Frederick is the author of more than 50 research publications, 4 book chapters and over 60 regional, national and international conference presentations on a wide range of topics in human factors and psychology. She is active in a number of professional associations, and is a Consultant for Psi Chi, the National Honor Society in Psychology.

Dr. Li Ding, Embry-Riddle Aeronautical University

Li Ding is a visiting professor of the Department of Engineering Fundamentals at Embry-Riddle Aeronautical University, where she has been since 2012. She received her Ph.D in Environmental Engineering from the University of Illinois at Urbana-Champaign in 2010. She taught several undergraduate courses in engineering and in science, and she currently teach Introductory to Programming for Engineers. From a background of an engineer, she is transitioning into an educator, and has been working with other principle researchers on education studies since 2015.

Rebecca Rohmeyer, Embry-Riddle Aeronautical University

The Application of Second Language Acquisition to Programming Language Study in a Blended Learning Environment

Abstract

This paper describes a design and implementation of a Second Language Acquisition in a Blended Learning (SLA-aBLe) project that aims to examine the efficacy of SLA approaches for teaching programming language. The project, which has been running for three semesters, modifies specific learning modules in a programming language class using a series of shorter videos with subtitles, online quizzes with tiered questions and comments, and a topic specified discussion board with Q&A sections. The SLA aspect of the SLA-aBLe study is emphasized through the use of strategies defined as best-practice SLA techniques, such as the inclusion of self-testing tired questions and visual-aided explanation in screencasts, more online programming writing assessment, more collaboration, and 'speak aloud' in labs. A series of surveys assessing students' perceptions, attitudes, and satisfaction of students in the SLA-aBLe, and control groups were analyzed. Their academic performance on exam scores was compared. A random group of students were selected and interviewed face-to-face each semester to understand the effectiveness of the SLA-aBLe design. Assessment results confirmed the effectiveness of SLA-aBLe design.

Introduction

Programming language is a common mandatory course taught in the first year of engineering and computer science programs. These types of courses typically utilize a common programming language (MATLAB, C, Java) to teach students about syntax and programming techniques and to introduce students to applied problem solving[1-4]. Learning a computer programming language has been known to be difficult for high-school and university students because of the lack of time for practice[5], in addition to the conceptual complexity of the topic and logical reasoning processes required for understanding. Programming courses are critical to the learning needs of students in STEM majors, as they provide students with problem solving skills that are easily transferrable and contextually relevant to math and science courses in the curriculum.

A programing language typically involves new vocabulary (keywords), punctuation (symbols), and grammatical structures (syntax) that people need to understand in order to communicate with computers[5-9]. In other words, a programming language is like a second language. Just as knowledge of the vocabulary, grammar, and punctuation do not make someone fluent in a spoken language, being a successful programmer requires more than just rote knowledge. Current introductory programming courses often struggle to provide enough problem solving because so much time is spent on learning the rote elements of the language[10].

By applying the well-developed cognitive frameworks used in second language acquisition (SLA) [11-15], a Blended Learning (aBLe) course was developed[16]. In this NSF funded project, different cognitive skills are focused at each of five stages of SLA with the implementation of associated instructional strategies in an Introduction to Computing for Engineers course at a private institution in the southeast[14]. The course teaches engineering students how to learn a programming language, MATLAB in a blended learning mode[17-24]. This paper describes the design, implementation of the project across three semesters. Discussion will

also focus on the continuous improvements in year two of the project based on the results and feedback obtained in year one[25-27].

SLA-aBLe Project Design

The project was started in summer 2015. Five topics (introduction to MATLAB, data type, input and output, conditional statement, and loop) were designed and implemented using techniques recommended in a SLA approach and aBLe environment. The blended learning environment is defined as a combination of the face to face and online learning environment to utilize strengths of both. Previous research showed that blended learning offers flexibility in terms of availability, and self-paced learning to the students[21-24]. The SLA approach divides learning into five stages, which are preproduction, early production, speech emergence, intermediate fluency, and advanced fluency. During each learning stage, best practices for teaching and learning are provided. This information and how it was applied in the SLA-aBLe project are presented in Table 1 below. More informative pictures, cartoons, tables, interactive tiered questions following Bloom's taxonomy, and MATLAB programming were included in the new learning materials, which were recorded at a slower speed of narration according to SLA[14]. The font of the learning materials was changed from an easy to read font, Calibri, to a hard-to-read font, Comic Sans MS so that the materials can improve memory performance and educational outcomes[28]. There were interactive questions embedded in the videos, which helped test students' understanding, and the videos could be watched as many times as students wanted. It is hoped by watching a series of short videos and answering tiered questions, students can achieve the preproduction stage as specified in SLA.

Table 1. A comparison of current blended learning and SLA-aBLe development

	Preproduction (minimal comprehension)	**Early Production (limited comprehension)**	**Speech Emergence (increased comprehension)**	**Intermediate Fluency (very good comprehension)**	**Advanced Fluency**
Current Blended Learning	Few pictures and visuals. Some topics are not well explained. Not enough self testing questions in the screencasts.	There are multiple choice questions but no simple programs. Facebook is used but there is no group discussion.	Students begin reading and writing in their programming language by solving different engineering problems.	Give students more challenging problems to synthetize what they have learned.	Open-ended engineering project to challenge their understanding and expand their knowledge.
Teaching Strategies in SLA-aBLe	Use pictures and visuals; speak slowly and use simple and shorter words to draw connection between SLA and programming languages; Reinforce learning by giving more self testing questions without adding in pressure.	Reinforce learning by asking students to produce simple programs in addition to the multiple choice questions; use discussion board to encourage group discussion.	Emphasize tiered questions and ask students to do a "think, pair, share" to process the new concepts.	Emphasize compare and contrast different concepts. Allow students to explain their problem solving process.	Project presentation opportunity will be offered to students to enhance their understanding .

Research Questions and Topics of Interest to be Presented

The information and the research questions that will be addressed in this paper include:

1. A discussion of course improvements made from Year 1 to Year 2 of the project
2. Results from the demographic, motivational and workload assessments used in the project
3. From the motivational aspect of assessment, we wished to answer the following research question:
   - I. Will SLA-aBLe help motivate students to learn in a simplified and easy to understand environment?
   - II. Will SLA-aBLe improve student performance in programming language study? This question was assessed by comparing student grades across SLA-aBLe and non-SLA sections of the course.
   - III. How did students perceive the effectiveness of their learning experience in the SLA-aBLe course?

1. Project Improvements in Year 2

During year one of the project, the researchers conducted a random prize drawing for students in the class who responded to the assessment surveys. At the time the students received their prize, they were also asked to participate in a short, voluntary interview conducted by the primary researcher to gain information about their perception of the class[25]. Based on students' feedback in year one, we continued to improve the project design in the second year by adding subtitles to each video to help understand the content; adding music at the beginning and the end of the video to create a relaxed study environment; reducing some video length to10 minutes long to keep their attention. Past research shows that at the preproduction stage of SLA, students have minimal comprehension[11-14]. They will try to comprehend the given messages than they produce. The instruction should be clear and easy to understand. Figure 1 shows a snapshot of new video design with subtitle.

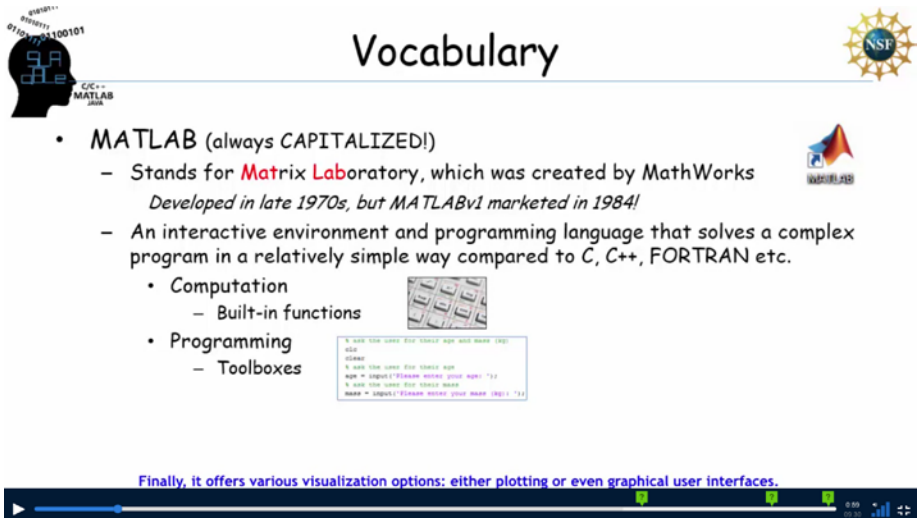


Figure 1. A snapshot of new video design with subtitle in the second year

Early production skills were obtained by asking students to take an online quiz after watching videos. Improvements included adding comments to quiz questions and answers to help understanding the mistakes and guide them to the right answers; changing completing the

whole program writing problem to completing the incomplete programming writing problem to reduce student's work load. A discussion board on Canvas was used to facilitate group discussion and provide instructional assistance online. Improvement included adding Q&A to the discussion board to answer common questions students have. Figure 2 shows new online quiz design with comments added in each answer. Figure 3 shows the new discussion board design with Q&A. On the second day in the lab, each instructor spent the first 5-10 minutes to go over the common mistakes found in the online quizzes. Then students were required to conduct "think, pair, share" exercises in the following 25 minutes so that they can think about what they have learned online, explain their learning to their partners, and share their experience facilitating cognitive skills development in the speech emergence stage. After the "think, pair, share" exercise, students were allowed to start their more complicated individual assignment. It is expected that after the completion of the individual assignment, students can demonstrate excellent comprehension and enter the intermediate fluency stage. Finally, at the advanced fluency stage, students develop and refine their knowledge of more sophisticated aspects of grammar and syntax when they start the open-ended final project. It is expected the final project can enhance student's understanding of the comprehensive materials learned in the whole semester.

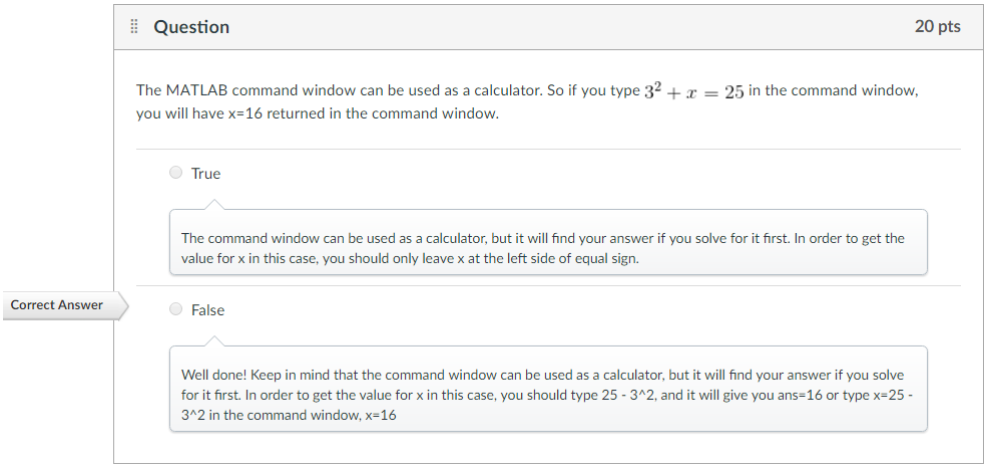
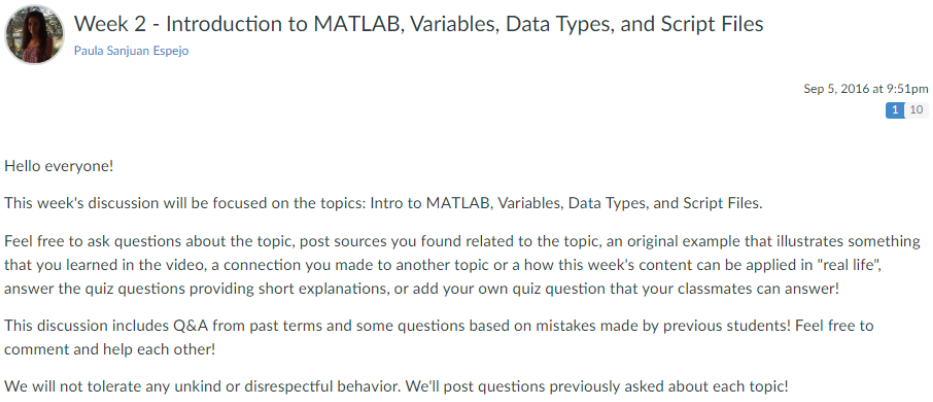


Figure 2. New online quiz design with comments added in the second year



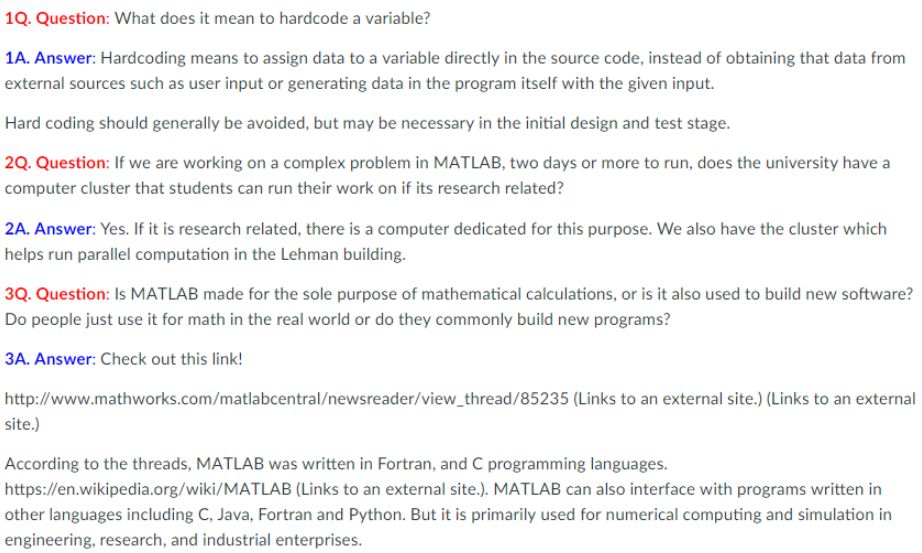


Figure 3. Redesigned discussion board with Q&A in the second year

2. Assessment Results

There were six surveys conducted in each semester, with three instructors each teaching at least one experimental (SLA-aBLe) section and one control (non-SLA-aBLe) section. A demographic survey was collected at the beginning of each semester to check student's foreign language and programming language experience. There were a total of 203 students in three semesters who completed the surveys with a response rate of 36%. Table 2 shows the descriptive statistics of student's language experience. Eighty-four percent (83%) of students chose English as their native/first language. When students were asked about their language experience, fifty percent (50%) of the students indicated that they do speak other languages.

Table 2. Descriptive statistics of student's language experience

Language	Not at all fluent (%)	Not very fluent (%)	Moderately fluent (%)	Somewhat fluent (%)	Very fluent (%)
English	0	2.41	1.2	9.64	77.11
Chinese	25	3.85	0	0	7.69
German	31.75	6.35	4.76	1.59	3.17
Spanish	11.7	25.53	24.47	4.26	11.70
Vietnamese	27.45	0	0	0	0
French	22.73	15.15	10.61	4.55	6.06
Arabic	25.93	0	1.85	0	5.56
Korean	32	4	0	0	2
Portuguese	26.92	7.69	1.92	0	0
Other	19.12	11.76	2.94	2.94	14.71

Table 3 indicates student's prior programming language experience. Since this is the entry-level programming course, the majority do not have previous programming language experience. 33.33% of the students confirmed their previous programming language experience as listed in Table 3. In addition, NASA TLX, a well-established measure of self-assessed workload was used to measure six workload subscales: mental demand, physical demand, temporal demand, performance, effort and frustration[32]. The NASA-TLX assumes that some combination of these subscales is likely to represent the workload experienced by most people

performing most tasks. The NASA-TLX was analyzed and the average results are shown in Figure 4.

Table 3. Descriptive statistics of student's programming language experience

Programming language	Low skill level (%)	Moderately low skill (%)	Moderate skill level (%)	Moderately high skill (%)	High skill level (%)
MATLAB	48.84	20.93	23.26	4.65	2.33
Fortran	96	0	4	0	0
Java	53.66	17.07	17.07	9.76	2.44
C/C++	73.53	11.76	14.71	0	0
Visual Basic	80	10	10	0	0
Python	58.14	16.28	20.93	4.65	0
Other	70	10	13.33	3.33	3.33

For the NASA TLX study, from Figure 4 we can see that students in SLA-aBLe sections reported lower workload demands than students in the non-SLA-aBLe sections except the physical demand, which can be contributed to more programming practices students had to accomplish during the SLA-aBLe online study. This explains the effectiveness of the SLA-aBLe design.

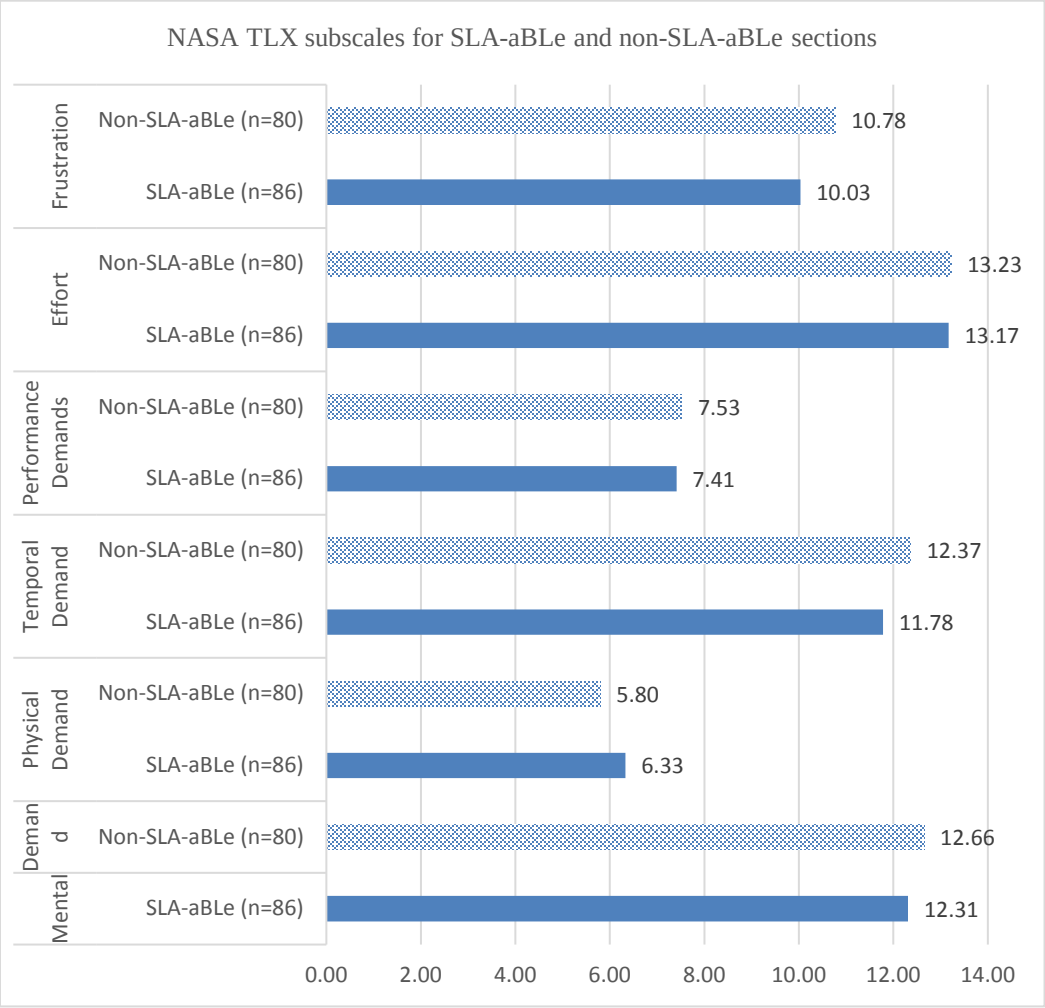


Figure 4. NASA TLX subscales for SLA-aBLe and non-SLA-aBLe sections

Research Question I: Motivation Differences

The Motivation Inventory (IMI) was used to answer the first research question. IMI assesses student's motivation across five subscales including interest/enjoyment, perceived competence, importance, felt pressure and tension, and perceived usefulness on a scale of 1 to 7, with 1 being "not true at all" and 7 being "very true". The IMI has been validated for use with college student populations[29-31]. To the IMI study, from five subscale scores in Figure 5 we can see that students showed less pressure and higher competence, enjoyment, usefulness, and importance in SLA-aBLe section than the non-SLA-aBLe students, which confirmed the positive study experience students received in SLA-aBLe sections.

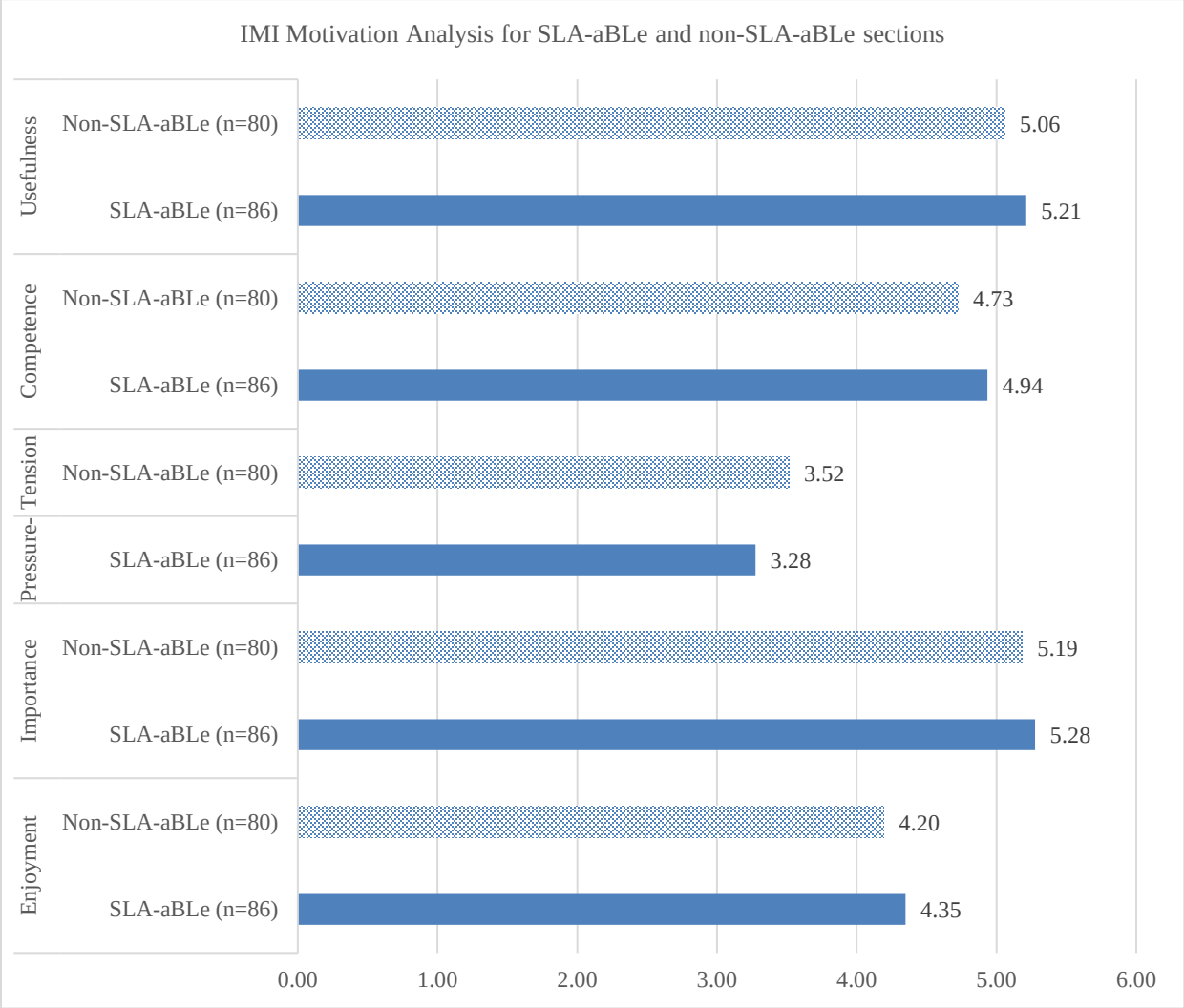


Figure 5. IMI Motivation Analysis for SLA-aBLe and non-SLA-aBLe sections

Research Question II: Student Performance Differences

The second research question was answered by running a chi-square test of independence on students' final grade in SLA-aBLe sections and non-SLA-aBLe sections for all three

semesters. There was no significant relationship associated between the course sections and final grade, however there were more A and B grades and less F grades in SLA-aBLe sections than those in non-SLA-aBLe section as shown in Figure 6.

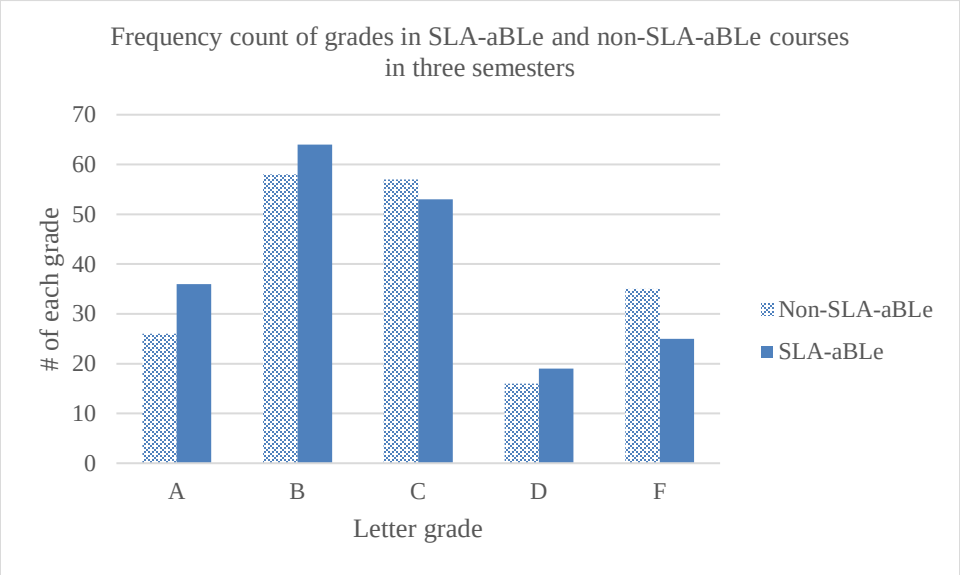


Figure 6. Frequency counts of grades in SLA-aBLe and non-SLA-aBLe sections in three semesters

Research Question III: Student Experience in the SLA-aBle course

The third research question was answered by analyzing face-to-face interview results. Six students each semester were interviewed regarding their perception of the course design and their experiences. The questions asked during the interview are listed in Table 5.

Table 5. Face-to-face interview questions in three semesters

Number	Questions
1	Please indicate your previous second language, and programming language experience.
2	Are you in the non-SLA-aBLe section? What is your biggest concern of the class?
3	If you are in the SLA-aBLe section, please answer the following questions: • Do you like the new videos? If yes, what do you like most? If no, explain. • Do you like the online quizzes? If yes, what do you like most? If no, explain. • Do you like the discussion board? If yes, what do you like most? If no, explain. • Do you like the think-pair-share in the lab? Please explain. • Does SLA-aBLe helped engage the study of programming language in a simplified and easy to understand environment? Please explain

From these interviews it was suggested that the biggest concern is the feeling of intimidation in learning a programming language. Students in the SLA-aBLe course sections believed that that teaching programming using SLA was helpful to their learning. Students who have a second language learning experience especially confirmed this during the interview. Students indicated more engagement with the online interactive video, compared to the topics that were presented in a traditional non-interactive format. The captions in the videos help students understand the specific terms. Music does not play an important role in the video design. They pointed out that the tiered examples in the videos and tiered quiz questions eased their anxiousness and helped their comprehension of the materials. Students expressed a desire to

flip all topics to SLA-aBLe format. Students also commented on the laboratory sessions, indicating that the "think, pair, share" activity encouraged the collaboration which was helpful to learning and comprehension. Students would rather take the discussion board as an open source information system than use it as an online discussion area.

Conclusion and Future Work

This paper presented a continuous study of the SLA-aBLe project in three semesters, which was started in the summer of 2015. The study tests the hypothesis that the use of cognitive frameworks in second language acquisition for the development of a blended learning experience of programming languages can improve engagement and the learning experience of engineering students. Quantitative and qualitative data were collected during the three-semester study. The first research question was answered by conducting IMI survey six times each semester to the students in SLA-aBLe and non-SLA-aBLe sections. For the IMI study, students showed less pressure and higher competence in SLA-aBLe section than the non-SLA-aBLe students. They reported higher level of enjoyment, usefulness, and importance before the end of course survey. The second research question was answered by running a chi-square test of independence on students' final grade in SLA-aBLe sections and non-SLA-aBLe sections in three semesters. There was no significant relationship between the course sections and final grade, however there were more A and B grades and less F grades in SLA-aBLe sections than those in non-SLA-aBLe section. The third research question was answered by analyzing face-to-face interviews in three semesters. From 18 interviews conducted, they all indicated effectiveness of SLA-aBLe design, which includes interactive videos with captions, tiered examples, and questions online, and collaborative learning in the lab. Positive results let researchers believe that SLA-aBLe is a promising approach. They will continue to examine and analyze the trend. It is the researchers' desire to apply SLA-aBLe to any programming language study to facilitate student learning experience.

Acknowledgements

The authors are grateful for the support provided by the National Science Foundation, Division of Engineering Education and Centers, grant number EEC 1441825. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors also would like to acknowledge the effort from Ms. Caroline Liron, Dr. Matthew Verleger, who helped conduct the project in their classes, Dr. James Pembridge who offered suggestions on the project design and implementation, and the support from the Institution Research at Embry-Riddle Aeronautical University who conducted and collected the survey data for this project.

Bibliography

1. Bualuan, R. (2006). Teaching Computer Programming Skills to First-year Engineering Students Using Fun Animation in MATLAB," Paper presented at the 2006 American Society for Engineering Education Annual Conference & Exposition, Chicago, IL.
2. Devnes, P.E. (1999). MATLAB and Freshman Engineering. Paper presented at the 1999 American Society for Engineering Education Annual Conference & Exposition, Charlotte, NC.
3. Morrell, D. (2007). Design of an Introductory MATLAB Course for Freshman Engineering Students. Paper presented at the 2007 American Society of Engineering Education Annual Conference & Exposition, Honolulu, HI.
4. Naraghi, M.H.N. & Litkouhi, B. (2001). An Effective Approach for Teaching Computer Programming to Freshman Engineering Students, Paper presented at the 2001 American Society for Engineering Education Annual Conference & Exposition, New York.

5. Solomon, J. (2004). Programming as a Second Language. Learning & Leading with Technology, 32(4), 34-39.
6. Tran, L. (2014) Computer Programming Could Soon Be Considered a Foreign Language in One State. Retrieved March 7, 2014, from http://www.policymic.com/articles/81067/computer-programming-could-soon-be-considered-a-foreign-language-in-one-state
7. Tyre, P. (2013) Is Coding the New Second Language? Retrieved March 7, 2014, from http://www.smithsonianmag.com/innovation/is-coding-the-new-second-language-81708064/
8. Van Roy, P., (2003). The Role of Language Paradigms in Teaching Programming. Paper presented at the 34th SIGCSE Technical Symposium on Computer Science Education, New York, NY.
9. Wynn, M., (2015). Ky. Ponders Teaching Computer Code as Foreign Language. Retrieved January 29, 2015, from http://www.usatoday.com/story/tech/2015/01/29/ky-computer-code-as-foreign-language/22529629/
10. Victor, B. (2012). Learnable Programming. Retrieved March, 7, 2014, from http://worrydream.com/LearnableProgramming
11. Ellis, R. (1994). The Study of Second Language Acquisition. Oxford: Oxford University Press.
12. Krashen, S.D. (1981). Second Language Acquisition and Second Language Learning. Oxford: Pergamon Press.
13. Krashen, S. D. (1982). Principles and practice in second language acquisition. Oxford: Pergamon Press.
14. Krashen, S. D. & Terrell, T. (1983). The Natural Approach: Language Acquisition in the Classroom. London: Prentice Hall Europe.
15. Williams, J. (1999). Memory, Attention and Inductive Learning. Studies in Second Language Acquisition. 21: 1-48.
16. Marsh, D. (2012). Blended Learning Creating Learning Opportunities for Language Learners. Cambridge University Press.
17. Azemi, A., Pauley, L.L. (2006). Teaching the Introductory Computer-Programming Course for Engineering Using MATLAB and Some Exposure to C. Paper presented at the 2006 American Society for Engineering Education Annual Conference & Exposition, Chicago, IL.
18. Bjedov, G. & Andersen, P. (1996). Should Freshman Engineering Students be Taught a Programming Language. Paper presented at the Proceedings of the 26th Annual Frontiers in Education Conference, Salt Lake City, UT.
19. Herniter, M.E. & Scott, D.S. (2001). Teaching Programming Skills with MATLAB. Paper presented at the 2001 American Society of Engineering Education Annual Conference & Exposition, New York.
20. Abbitt, J. & Carroll, B. (2013). Using Technology to Enhance Undergraduate Learning in Large Engineering Classes. Paper presented at the 2013 American Society for Engineering Education Annual Conference & Exposition, Atlanta, GA.
21. Brown, C. & Meyers, D. (2007). Experimental Hybrid Courses that Combine Online Content Delivery with Face-to-face Collaborative Problem Solving. Paper presented at the 2007 American Society of Engineering Education Annual Conference & Exposition, Honolulu, HI.
22. Brown, C., Lu, Y., Meyer, D., & Johnson, M. (2008). Hybrid Content Delivery: Online lectures and Interactive Lab Assignments. Paper presented at the 2008 American Society for Engineering Education Annual Conference & Exposition, Pittsburgh, PA.
23. Yale, M., Bennett, D., Brown, C., Zhu, G., & Lu, Y. (2009). Hybrid Content Delivery and Learning Styles in a Computer Programming Course. Paper presented at the 39th ASEE/IEEE Frontiers in Education Conference, San Antonio, TX.
24. Sun, L., Kindy, M., & Liron, C. (2012). Hybrid Course Design: Leading a New Direction in Learning Programming Languages Paper presented at the 2013 American Society of Engineering Education Annual Conference & Exposition, San Antonio, TX.
25. Sun, L., Frederick, C., Espejo, P. S., & Cunningham, R. M., "Can We Teach a Programming Language as a Second Language?", Computer in Education Journal, Vol 7. 7 No. 3, pp105, 2016
26. Frederick, C., Sun, L., Liron, C., Verleger, M. Cunningham, R.M., & Espejo, P. S., "Implementation and Evaluation of a Second Language Acquisition – Based Programming Course", ASEE Annual Conference, New Orleans, LA, June 26-29, 2016
27. Cunningham, R. Sanjuan E. P., Frederick, C., Sun, L. & Ding, L. "A Second Language Acquisition Approach to Learning Programming Languages", ASEE SE Section Annual Conference, Tuscaloosa, AL, March 13-15, 2016
28. Diemand-Yauman, C. Oppenheimer, D., and Vaughan E. (2011) Fortune favors the bold (and the italicized): effects of disfluency on educational outcomes, Cognition, Vol 118 (1), pp 111-115.
29. Cordova, D. I., & Lepper, M. R. (1996). Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. Journal of Educational Psychology, 88, 715-740.

30. Dev, P. C. (1997). Intrinsic motivation and academic achievement. Remedial and Special Education, 18(10), 12-19.
31. Garcia, T., & Pintrich, P. R. (1996). The effects of autonomy on motivation and performance in the college classroom. Contemporary Educational Psychology, 21, 477-486.
32. NASA TLX (2014). NASA Task Load Index v 1.0. Human Performance Research Group: NASA Ames Research Center. Retrieved March 14, 2014, from http://humansystems.arc.nasa.gov/groups/TLX/downloads/TLX_pappen_manual.pdf.