Can We Teach a Programming Language as a Second Language?

Dr. Lulu Sun, Embry-Riddle Aeronautical University - 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 B.S. degree in Mechanical Engineering from Harbin Engineering University (China), in 1999, and 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 professional member of the Society of Fire Protection Engineers, and a member of the American Society for Engineering Education.

Dr. Christina Frederick, Embry-Riddle Aeronautical University - 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.

Miss Paula Sanjuan Espejo, Embry-Riddle Aeronautical University - Daytona Beach

I am an UG Aerospace Engineering student at Embry-Riddle Aeronautical University, Daytona Beach. I am from Spain and I am currently working on the SLA-aBLE project, the Implementation and Evaluation of Second Language Acquisition applied to programming courses.

Rachel Marie Cunningham, Embry-Riddle Aeronautical University

Rachel is a Graduate Research Assistant at ERAU in the Game-based Education & Advanced Simulations Lab. She has been historian for the ERAU Student Chapter of Human Factors & Ergonomics Society and a Psi Chi National Honors Society member for two years, and was Student Chair of the 10th annual Human Factors and Applied Psychology Conference, held on Embry-Riddle's campus. Her research interests are in design and creative thinking and along with this NSF grant project she is working on research for interpersonal development, which she has presented at two national conferences.

Can we teach a Programming Language as a Second Language?

Abstract

This paper describes a design and implementation of a NSF sponsored project in 2015. This study will test the hypothesis that the use of cognitive frameworks in second language acquisition for the development of a blended learning of programming languages can improve engagement and the learning experience of engineering students. Using this approach will place greater emphasis on problem solving techniques that can be utilized in all courses. The online module consists of a series of short videos (10-20 minutes), online quizzes with tiered questions, and topic specified discussion board led by student researchers. Students' demographic data, course-related behaviors such as usage of the instructional videos and discussion board, student performance such as quizzes and exams, and attitude toward the class will be compared across students in the experimental groups, and control groups to determine if student performance, behavior and attitudes vary across classrooms employing different teaching strategies.

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 computer[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 is developed that will accommodate a variety of learning needs and abilities, while potentially increasing student engagement in online components, reducing the intimidation and anxiety associated with learning programming languages, and providing better preparation for face-to-face classes[16]. SLA-aBLe will emphasize the problem solving needed in other general education courses instead of just keywords, syntax, and symbols. It will encourage the development of problem solving skills needed to persist in their higher education.

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

- Will SLA-aBLe help motivate students to learn in a simplified and easy to understand environment?
- Will SLA-aBLe improve student performance in programming language study?

- How does SLA-aBLe affect student problem solving ability?

SLA-aBLe Project Work

Learning a programming language is analogous to students acquiring a second language since it involves vocabulary, syntax, grammar and communicative outcomes as seen in a second language study. These skills must be sufficiently developed for the learner to function successfully in the environment that utilizes the language. In this project, different cognitive skills are focused on 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 the programming language, and MATLAB in a blended learning mode[17-24]. Table 1 shows a comparison of current blended learning and SLA-aBLe development. There are four topics (data type, input and output, conditional statement, and loop) which were designed in summer 2015 and implemented following the SLA approach in fall 2015. More helpful pictures, cartoons, tables, interactive tiered questions, 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 it can improve memory performance and educational outcomes[25]. There were three experimental classes (n=78) and four control classes (n=104) taught by three instructors respectively involved in the study in fall 2015.

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.

At each of the five stages of SLA, different proficiencies were focused on and different cognitive skills related to language learning were developed. PowerPoint slides were designed to include pictures, animation, self-analysis questions, and MATLAB code demonstration. After PowerPoint slides were designed, they were recorded into a series of 10 to 15 minutes long interactive screencasts. Figure 1 shows the snapshot of the PowerPoint slides and screencasts following SLA-aBLe development. Screencasts were uploaded to Edpuzzle website to track the usage statistics.

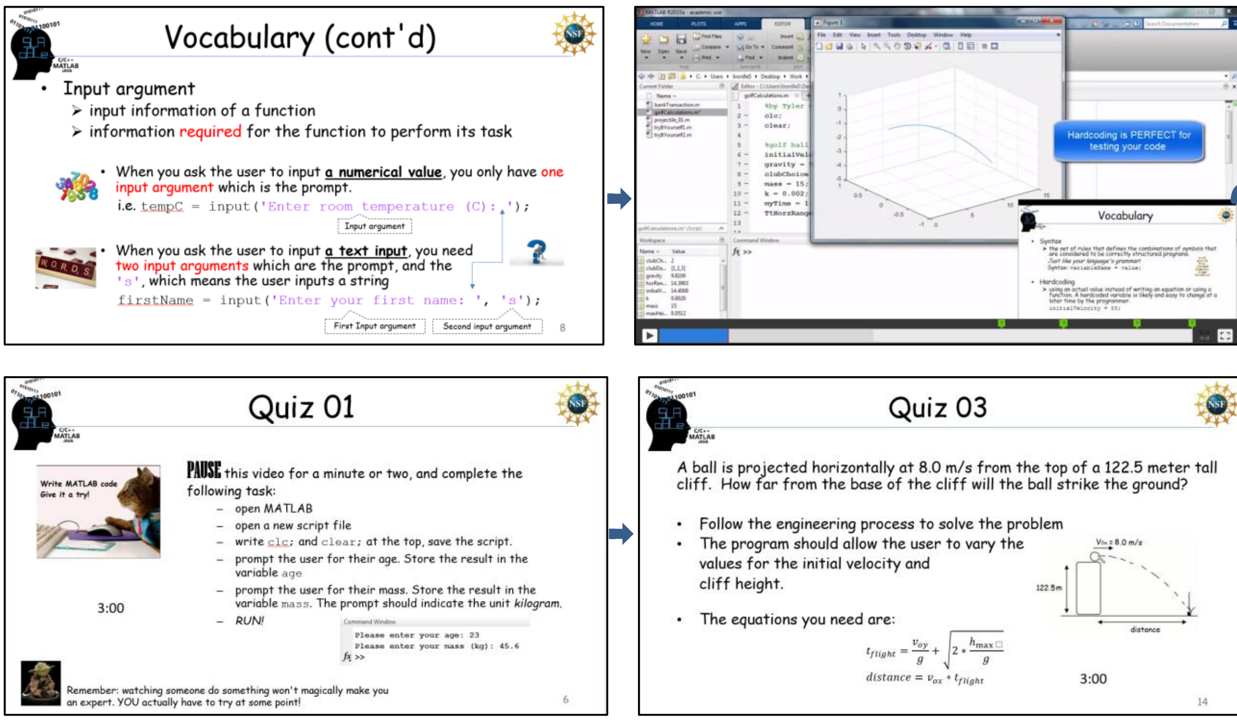
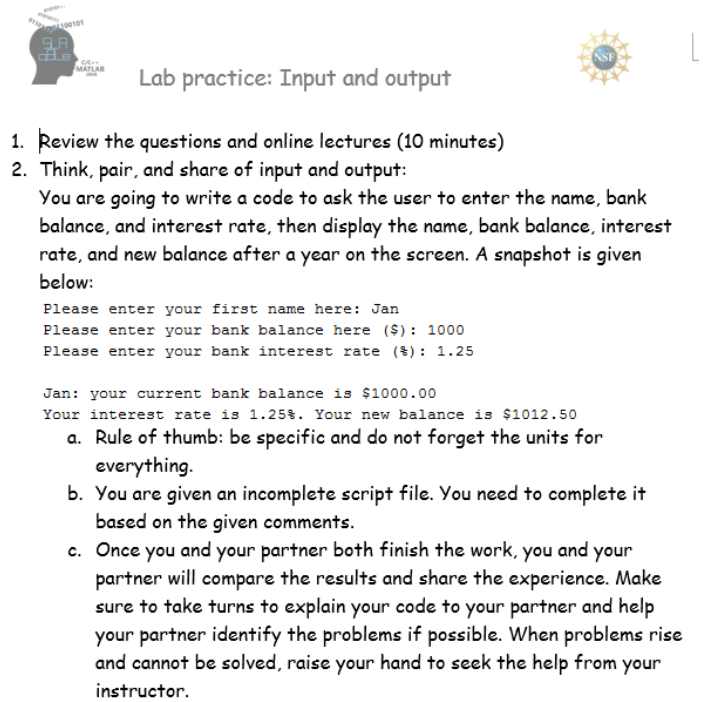


Figure 1. PowerPoint slides design following SLA-aBLe development

Early production skills were obtained by asking students to take an online quiz after each screencast study. There were usually five tiered questions in each online quiz. Students can take the quiz up to three times and the highest score was included into their gradebook. For each topic studied, there was at least one program writing problem included in the quiz which needed to be manually graded by research assistant and project researcher. A discussion board on Canvas was used to facilitate group discussion and provide instructional assistance online. 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. Figure 2 shows a snapshot of the “think, pair, share” exercise following SLA-aBLe development.


Lab practice: Input and output

1. Review the questions and online lectures (10 minutes)
2. Think, pair, and share of input and output:
   You are going to write a code to ask the user to enter the name, bank balance, and interest rate, then display the name, bank balance, interest rate, and new balance after a year on the screen. A snapshot is given below:

```
Please enter your first name here: Jan
Please enter your bank balance here ($): 1000
Please enter your bank interest rate (%): 1.25

Jan: your current bank balance is $1000.00
Your interest rate is 1.25%. Your new balance is $1012.50
```

   a. Rule of thumb: be specific and do not forget the units for everything.
   b. You are given an incomplete script file. You need to complete it based on the given comments.
   c. Once you and your partner both finish the work, you and your partner will compare the results and share the experience. Make sure to take turns to explain your code to your partner and help your partner identify the problems if possible. When problems rise and cannot be solved, raise your hand to seek the help from your instructor.

Figure 2. “Think, pair, share” implementation in the class time

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.

Assessment

There were seven surveys conducted in the fall 2015. A demographic survey was collected at the beginning of the semester. In addition, two measures were administered six times across the study to answer the first and the third research question. The first measure, the Intrinsic Motivation Inventory (IMI) was used to assess student’s motivation across five dimensions including interest/enjoyment, perceived competence, importance, felt pressure and tension, and perceived usefulness. The IMI has been validated for use with college student populations. The second measure, NASA TLX, a well-established measure of self-assessed workload was used to measure six workload dimensions: mental demand, physical demand, temporal demand, performance, effort and frustration. Student’s final grades were collected to examine the second research question.

The perceived workload and motivation were analyzed by running the t-tests and the results are shown in Table 2. There was only one mean difference in perceived workload found across six survey administrations. After viewing the input/output materials, students in the SLA-aBLe sections reported significantly lower frustration than students in the non-SLA-aBLe sections. This finding was consistent for all six types of workload: physical, mental, temporal, performance, effort and frustration. Additionally, at the end of the course, the perceived workload demands of the course were perceived to be lower overall in SLA-aBLe sections than those in non-SLA-aBLe sections, with the exception of mental workload. Results are highlighted in Table 2.

Motivational differences were found between students in SLA-aBLe course sections and students in non-SLA sections. After viewing the data types' materials, students in the SLA-aBLe section reported significantly higher levels of enjoyment, competence and usefulness for class information than students in non-SLA sections. In addition, students in the SLA-aBLe sections reported significantly lower levels of frustration than the non-SLA students after the data types topic was presented. After viewing the specialized input/output materials, students in the SLA-aBLe sections also reported significantly higher levels of usefulness for those materials than students in the non-SLA sections.

Table 2 Means for Workload and Motivation Variables across Administration Periods

		Administration Period					
		Week 1 of Course	Data Types	Input / Output	Conditional Statements	Loops	End of Course
Workload Variables	Class Section	Means					
Mental Demand	SLA (n=11) Non-SLA (n=10)	10.52 10.19	12.12 13.52	11.08 13.57	12.92 13.00	14.15 13.24	16.78 16.82
Physical Demand	SLA (n=11) Non-SLA (n=10)	6.00 5.38	5.96 7.29	6.67 6.43	6.17 5.62	7.19 6.53	8.44 12.45
Temporal Demand	SLA (n=11) Non-SLA (n=10)	10.64 8.38	11.44 11.90	8.25 11.21	10.67 10.92	10.38 11.94	17.33 16.18
Performance Demands	SLA (n=11) Non-SLA (n=10)	7.33 7.78	7.04 8.95	8.83 5.43	7.42 7.23	8.50 9.00	5.56 8.55
Effort	SLA (n=11) Non-SLA (n=10)	11.91 11.32	12.60 13.38	11.50 14.36	13.12 13.33	14.31 13.41	16.78 17.00
Frustration	SLA (n=11) Non-SLA (n=10)	8.45 8.32	8.44 11.52	8.42 13.00*	7.67 11.77	12.56 11.47	14.11 14.82
Motivation Variables							
Enjoyment	SLA (n=11) Non-SLA (n=10)	4.61 4.31	4.77 4.02*	4.82 4.41	4.64 4.49	4.23 4.01	4.27 3.90
Importance	SLA (n=11) Non-SLA (n=10)	5.23 4.73	5.42 4.98	5.72 5.23	5.62 5.12	5.65 5.02	5.98 5.78
Pressure-Tension	SLA (n=11) Non-SLA (n=10)	3.04 2.74	2.78 3.62*	2.71 3.69	2.40 3.32	3.95 3.19	4.30 4.62
Competence	SLA (n=11) Non-SLA (n=10)	4.76 4.98	5.05 4.20*	4.94 4.81	4.94 5.03	4.40 4.09	4.70 4.37
Usefulness	SLA (n=11) Non-SLA (n=10)	5.20 4.89	5.72 4.65**	5.85 4.93*	5.85 5.62	5.41 5.07	4.85 4.61

*= p<.05
**=p<.01

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. There was no significant relationship associated between the course section and final grade ($X^2(4) = 2.660$; p = .616). Students within the SLA-aBLe sections did not score higher in the class than students in the non-SLA-aBLe sections. Figure 3. Shows the frequency count of grades in SLA-aBLe and non-SLA-aBLe sections.

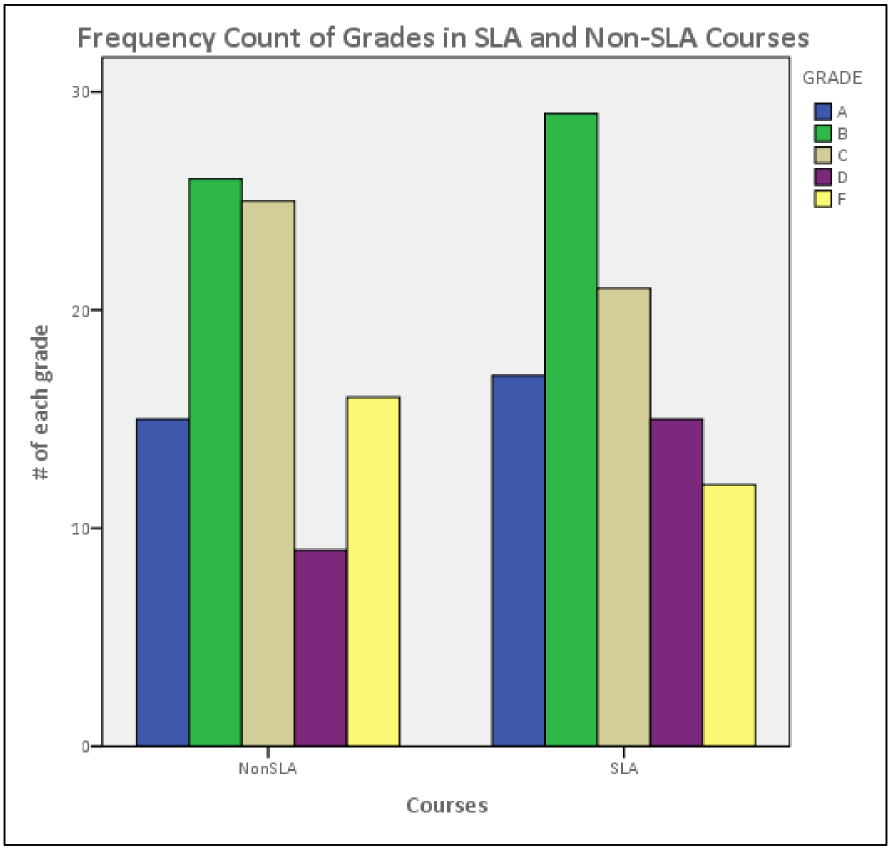


Figure 3. Comparison of students' final grades in the SLA-aBLe and non-SLA-aBLe sections for fall 2015

While these results from fall 2015 do not show significant differences, students in the SLA-aBLe sections did receive more A's and B's and fewer F's in the class than did non-SLA section students. This trend will be examed continuously in the future semesters.

Six students from the SLA-aBLe course sections were randomly selected and interviewed for feedback regarding the course design. From these interviews it was suggested that the SLA-aBLe course sections were effectively designed. They believed that teaching programming using SLA was helpful to their learning. Students indicated more engagement with the online video, compared to the topics that were presented in a traditional non-interactive format. 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 were not in favor of the online discussion board. They considered it as a work overload rather than an online communication tool. These results are consistent with the satisfaction survey results which students completed at the end of the semester as shown in Figure 4 and Figure 5 (n=19).

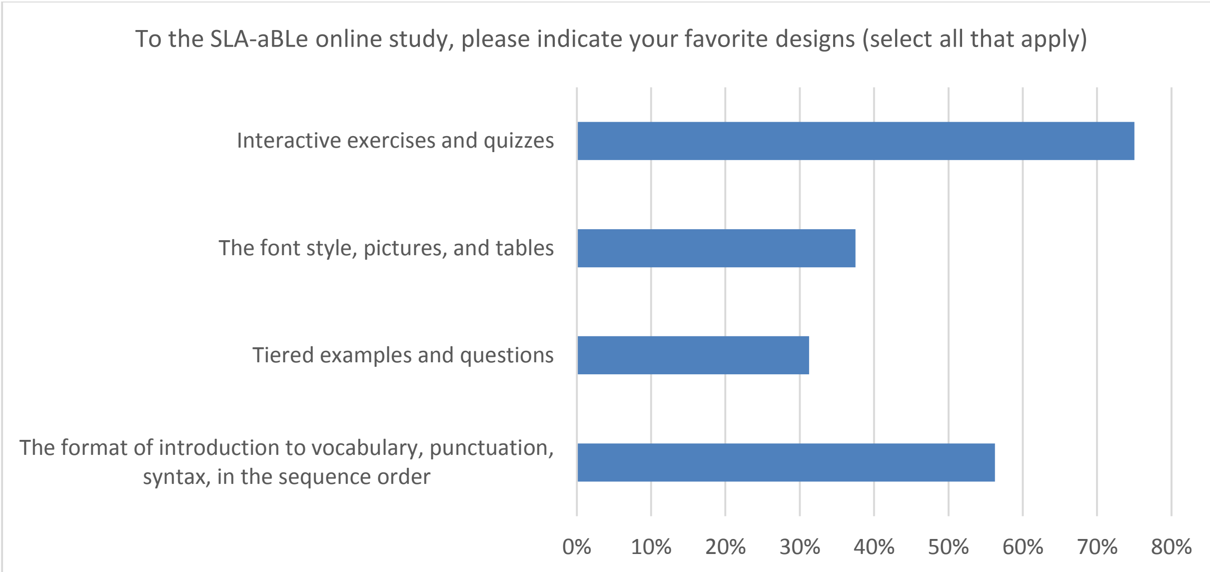


Figure 4. Satisfaction survey about the SLA-aBLe video design at the end of fall 2015

Questions	**Strongly Disagree**	**Disagree**	**Agree**	**Strongly Agree**	**Total**	**Weighted Average**
"Think, pair, share" enhanced understanding of the content	15.79%	5.26%	52.63%	26.32%	19	2.89
The online discussion board facilitated the online study	26.32%	47.37%	21.05%	5.26%	19	2.05
The program writing problem in the quiz helped test understanding the study materials	15.79%	10.53%	42.11%	31.58%	19	2.89
SLA-aBLe project helped engage the study of programming language in a simplified and easy to understand environment	5.56%	16.67%	44.44%	33.33%	18	3.06

Figure 5. Satisfaction survey about SLA-aBLe overall design at the end of fall 2015

Conclusion and Future Work

The SLA-aBLe project was designed in the summer of 2015 and implemented in the fall of 2015. This 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 study. The preliminary results are promising, but need further investigation. From the IMI and NASA TLX data, it was found that students' mean scores for perceived frustration in SLA-aBLe section were lower than those in non-SLA-aBLe sections except two (week 1 and after the loops video). At the end of the course, the perceived workload demands of the course were perceived to be lower overall in SLA-aBLe sections than those in non-SLA-aBLe sections with the exception of mental workload. While these differences were not statistically significant, they are interesting and may be important. This trend will be continuously examed during the future semester. Motivational differences favoring the SLA-aBLe students were shown after students viewed the data types' materials and the input/output materials. Specifically SLA-aBLe students reported finding the specialized materials they used as valuable, and for the data types' week they also reported higher enjoyment and competence and lower pressure. No differences were shown during the pre-test, during the presentation of conditional statements or loops, or at the end of the course. These results are consistent with the interview results and the satisfaction results at the end of the semester. These results are promising, but needs further investigation due to the small sample size used in this data collection. Researchers will continue to develop solutions to increase participation rates during future semesters.

Students' final grades from SLA-aBLe sections and non-SLA-aBLe sections were compared. Although there were no significant differences across the sections, there were more 'A', 'B' grades and less 'F' grades in SLA-aBLe sections than those in the non-SLA-aBLe sections. These results should be viewed cautiously and researchers will continue to examine end of course grades as one measure of learning effectiveness.

In spring of 2016, three same instructors and eight sections are being involved in the study (3 SLA-aBLe sections, and 5 non-SLA-aBLe sections). The project will be implemented in 3 SLA-aBLe sections and surveys will be implemented in SLA-aBLe and non-SLA-aBLe sections. The researchers will continue to examine and analyze the trend. It is the researchers' desire to disseminate the course modules to students and instructors who are either learning or teaching an introductory programming course to facilitate student learning outcomes.

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, Dr. Li Ding who helped conduct the project in their classes, 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. 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.