Nikou, S.A. & Economides, A.A. (2014). Transition in Student Motivation during a Scratch and an App Inventor course, In Proceedings of Global Engineering Education Conference (EDUCON), 2014 IEEE, 3-5 April 2014, Istanbul, Turkey, pp. 1042 - 1045,DOI > 10.1109/EDUCON.2014.6826234

Transition in Student Motivation during a Scratch and an App Inventor course

Stavros A. Nikou
Interdepartmental Program
of Postgraduate Studies in Information Systems
University of Macedonia
Egnatia Street 156, 540 06, Thessaloniki, Greece
+30-2310-891768
stavrosnikou@sch.gr

Anastasios A. Economides
Interdepartmental Program
of Postgraduate Studies in Information Systems
University of Macedonia
Egnatia Street 156, 540 06, Thessaloniki, Greece
+30-2310-891768
economid@uom.gr

Abstract— **Considering the declining enrolling in computing fields and the increasing demand in STEM disciplines, innovative methods should be employed to attract students in computing disciplines. MIT Scratch and App Inventor for Android visual programming environments are two such approaches. This is a comparative study to investigate any differences in the transition of students' motivation to learn programming using Scratch and App Inventor for Android in K-12 educational settings. Intrinsic goal orientation, task value, control of learning beliefs and self efficacy were found to be increased using these two entry-level learning programming environments from the beginning to the middle of the course. No effect on extrinsic motivation was found. Evaluating the transition in motivation throughout the whole course period for both environments (work in progress) will have an impact on educators to retain students' interest in programming and improve their attitudes towards computing.**

Keywords— computer science education, programming, curriculum, motivation, computer languages, AppInventor, Scratch

I. INTRODUCTION

Due to the increasing diffusion of technology in all aspects of our living, we are facing an increasing demand in STEM (Science, Technology, Engineering and Mathematics) disciplines. According to the American Bureau of Labor Statistics, there will be 9.2 million jobs in STEM fields in the US, by 2020. Half of those jobs will be in computing or information technology [1]. According to Cedefop, the European Centre for the Development of Vocational Training, there will be a growing demand in STEM professionals in Europe, especially in the ICT field, for the following years [2]. However, despite the growing demand on high skilled computing professionals, there is a diminished interest in computing and a decreasing enrollment in CS majors [3]. It is of great importance to excite student's interest about STEM and computing careers from their early stages in education. Computing is not only an important discipline itself in our today's ICT driven society but it is integral to all engineering disciplines. Therefore all students need to be motivated in order to learn the basic computer science principles. It is quite difficult though to trigger and maintain student motivation to learn how to program. A variety of programming environments have been used to teach entry level programming ([4], [5], [6]). Some of them e.g., Alice, Scratch, Greenfoot, Lego Mindstorms, App Inventor for Android, manage to engage students and improve their attitudes towards Computer Science and Computer Engineering ([7], [8], [9], [10], [11]).

The present study examines the effect of two of these entry-level learning programming environments, namely Scratch and App Inventor for Android (AIA) on the transition of K-12 students' motivation to learn programming during a course period. Understanding how students' motivation progresses throughout the course could help educators to determine how to best utilize the two programming environments in the classroom in order to retain students' motivation.

I. BACKGROUND

A. Scratch

Scratch is an educational visual programming environment developed by the MIT Lifelong Kindergarten group. Since its initial release in 2003, educators all over the world have been using it successfully to teach basic CS concepts. Scratch is a programming environment that offers great support to spark interest among young students towards programming ([12], [13], [14], [15], [16]). It allows program creation by embedding programming blocks in a visual drag and drop programming environment with built-in graphics creation and sound editing capabilities. It supports novice users and especially youngsters to create animated stories, multimedia presentations, games, simulations and other interactive projects. Students also have the option to share their creations on the web through the Scratch online community. New programmers, working on meaningful projects, learn programming, computational thinking, reasoning creatively and working collaboratively.

B. App Inventor for Android

App Inventor for Android (AIA) is a relatively new visual educational programming environment for creating Android mobile apps by combining programming building blocks. It was originally part of a Google pilot program and now it is maintained by the MIT Center for Mobile Learning. AIA has two main components: a designer window where the student builds the user interface and a blocks editor window where the application logic and behavior can be programmed. The

application can be tested on a PC emulator or directly to an Android mobile phone. AIA can efficiently support introductory level courses at K-12 and college levels ([17], [18], [19]).

The authors have chosen Scratch and AIA in order to examine their effects on student's motivation (as being promising to engage students) because they share the same basic programming interfaces (drag enabled programming building blocks) and pedagogical principles. They are both object oriented and event driven environments with a low threshold entry to programming. They are both based on constructionist learning theories where learning is treated as a process of knowledge reconstruction by the individuals rather than as a simple knowledge transmission from the instructor [20].

II. METHOD

A. Participants

The participants are 38 students, 19 males (50%) and 19 females (50%), enrolled in an introductory programming course, in a Greek senior-high school. The average age of the students is 16.4 years old. All students have had a very little experience in programming· they all had taken a computing class, two years ago, where they had a very short exposure to primitive programming principles (e.g. sequential execution of simple commands to draw simple geometrical shapes) using a Logo-based environment.

The participants were randomly assigned into two groups (classes). The control group consisted of 18 students, 8 males (44.5%) and 10 females (55.5%) and the experimental group consisted of 20 students, 11 males (55%) and 9 females (45%). In the control group (the Scratch class) the programming environment used is Scratch and in the experimental group (the AIA class) is AIA. Both groups encountered Scratch and AIA for the first time.

B. Course description

The course is an introduction to programming for non-major senior high school students. The classes have the same instructor (a CS educator expert) and are met twice a week. Programming concepts such as sequential execution, concurrency, event handling, Boolean logic, conditional statements, iteration-looping, variables, procedures, user interface design, sound and sensors will be taught to both classes with the same sequence whenever possible. A constructionist approach with learn-by-making or learn-by-doing practices is the pedagogical underpinnings under the programming environments used [20].

Our instructional approach is Problem Based Learning (PBL). In PBL students try to solve authentic problems with an inquiry approach ([21], [22]). Under the PBL approach, students develop problem-solving and self directed learning abilities [23] while their learning motivation is increased [24]. Before the beginning of the course we have conducted a diagnostic test about programming in both groups in order to evaluate any pre-existing programming knowledge. Similar formative assessments are conducting during the classes. After completing the basic principles, towards the end of the course, students will work on a final project and a final assessment will follow. All assessments were designed by the course instructor.

C. Motivated Strategies for Learning

In order to measure the motivation of students, we used the Motivated Strategies for Learning Questionnaire (MSLQ) developed by Pintrich, Smith, Garcia, and McKeachie [25], a widely used instrument for investigating self regulated learning strategies. Two sections comprise the MSLQ, a motivational section and a learning strategies section. The present study considers only the two of the three dimensions of the motivational section: the value components (intrinsic goal orientation, extrinsic goal orientation, task value) and the expectancy components (control-beliefs, self-efficacy).

The MSLQ uses Likert-type scale questions (1 = *strongly disagree*, to 5 = *strongly agree*) to measure students' motivational orientations. Since self regulated learning is context specific [25] and our purpose is to measure motivation in the specific context of learning programming the survey questionnaire is directly defined in the context of specific subject (programming). For example, a self-efficacy item is "I believe that I am able to learn well how to program"; an intrinsic motivation item is "Even when completing the assignments does not guarantee that I get a good grade, I still love to complete them".

D. Research design

In order to evaluate the impact of Scratch and AIA on students' motivation we implemented the procedure depicted in Fig.1.

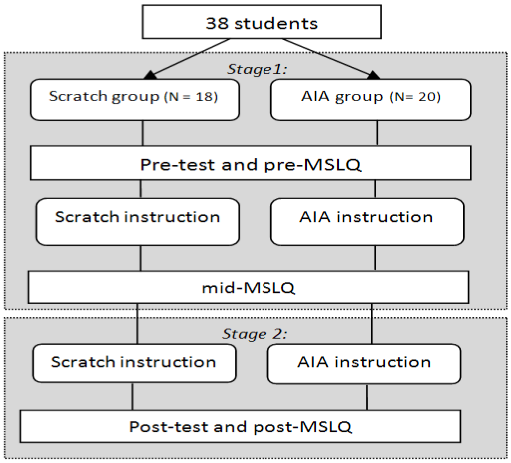

Fig. 1 Research Procedure

Stage 1 has been implemented so far and stage 2 is in progress. The MSLQ was administered two times during stage 1: once before students got introduced to the new programming environments at the beginning of the course (pre-MSLQ) and one more time after 7 weeks (mid-MSLQ). The pre-MSLQ reveals pre-existing level of motivation towards programming before students get introduced to Scratch and AIA and the mid-MSLQ reveals the motivational levels in the middle of the course respectively. It is scheduled to be administered one more time during Stage 2, after 8 weeks, when the course finishes along with the final exam

(post-MSLQ). These three motivational snapshots will give us the opportunity to examine motivation progression throughout the course and compare the motivation between the two groups (Scratch group and AIA group). Before the beginning of the course we have conducted a diagnostic test about programming in both groups (pre-test) in order to evaluate any pre-existing programming knowledge. After completing the basic principles, towards the end of the course, students will work on a final project and a final assessment will follow (post-test) to evaluate course performance. Student performance data will be evaluated to investigate any impact that the programming environments have on student achievement in relation to students' motivational levels.

III. DATA ANALYSIS

Before the beginning of the course, ANOVA tests showed no significant difference, at the $p < 0.05$ level, for both the Scratch and the AIA groups on intrinsic goal orientation ($F(1,36) = 0.118$, $p = 0.731$), extrinsic goal orientation ($F(1,36) = 0.001$, $p = 0.976$), task value ($F(1,36) = 0.076$, $p = 0.785$), control of learning beliefs ($F(1,36) = 0.593$, $p = 0.446$) and self-efficacy ($F(1,36) = 0.004$, $p = 0.950$).

The internal reliability coefficients for the MSLQ subscales for both groups for the pre-test as well as the mid-test were all satisfactory. Table 1 shows estimated Cronbach's alpha values for all subscales for both tests.

TABLE I. RELIABILITY ANALYSIS USING CRONBACH ALPHA MEASURE

Scales	Pre-test		Mid-test	
	Scratch group N = 18	AIA group N = 20	Scratch group N = 18	AIA group N = 20
Intrinsic Goal	0.74	0.77	0.75	0.81
Extrinsic Goal	0.68	0.71	0.63	0.69
Task Value	0.85	0.87	0.84	0.88
Control of Learning Beliefs	0.69	0.71	0.67	0.70
Self-Efficacy	0.78	0.71	0.76	0.83

Mean values and standard deviations as well as t-tests for each motivational subscale were estimated in order to evaluate the impact of the Scratch programming environment upon student motivation. Table 2 shows mean values, standard deviations and t-tests for the five motivational sub-scales for pre-test and mid-test for the Scratch group. Significant increase was found regarding intrinsic goal orientation, task value, control of learning beliefs and self-efficacy (Cohen's d effect sizes were respectively $d = 0.56$, $d = 0.69$, $d = 0.17$ and $d = 0.58$). No significant difference was found for extrinsic goal orientation.

Mean values and standard deviations as well as t-tests for each motivational subscale were estimated in order to evaluate the impact of the AIA programming environment upon student motivation.

TABLE II. T-TESTS FOR THE FIVE MSLQ SUBSCALES FOR THE SCRATCH GROUP (N = 18)

Scales	Mean (SD)		df	t
	Pre-MSLQ	Mid-MSLQ		
Intrinsic Goal	3.76 (0.67)	4.20 (0.87)	17	3.80**
Extrinsic Goal	4.04 (0.45)	4.05 (0.42)	17	0.35
Task Value	3.71 (0.66)	4.24 (0.85)	17	3.29**
Control of Learning Beliefs	3.62 (0.98)	3.79 (1.00)	17	2.35*
Self-Efficacy	3.73 (0.69)	4.16 (0.79)	17	4.99**

* $p < 0.05$, ** $p < 0.01$

Table 3 shows mean values, standard deviations and t-tests for the five motivational sub-scales for pre-test and mid-test for the AIA group. There is a positive impact on motivation for the intrinsic goal orientation ($d = 0.94$), the task value ($d = 1.45$), control of learning beliefs ($d = 0.16$) and self-efficacy ($d = 0.61$) subscales for the AIA group. No significant difference was found for extrinsic goal orientation.

TABLE III. T-TESTS FOR THE FIVE MSLQ SUBSCALES FOR THE AIA GROUP (N = 20)

Scales	Mean (SD)		df	t
	Pre-MSLQ	Mid-MSLQ		
Intrinsic Goal	3.69 (0.61)	4.26 (0.59)	19	4.85**
Extrinsic Goal	4.04 (0.75)	4.08 (0.74)	19	1.32
Task Value	3.64 (0.79)	4.92 (0.96)	19	6.61**
Control of Learning Beliefs	3.49 (0.87)	3.63 (0.85)	19	2.82*
Self-Efficacy	3.72 (0.59)	4.11 (0.68)	19	4.67**

* $p < 0.05$, ** $p < 0.01$

IV. DISCUSSIONS AND FUTURE WORK

Scratch's main feature is that it is appealing to people with no programming experience at all [15]. AIA is considered as a Scratch descended for mobile phones. It offers mobile application creation that "facilitates interactions with the world outside of the classroom" [26]. This study examines how student motivation progresses throughout a Scratch and an AIA course. Previous research [27] has shown that the change of the motivation in a programming course from early to middle was more remarkable than the change from middle to late in the course. The present study has collected so far (beginning to middle of the course period) the following evidences regarding student motivation for the Scratch and AIA groups:

1) There is an increase in intrinsic goal orientation for both groups. The motivation to engage in programming arises from intrinsic rewards (challenge, curiosity, mastery). Intrinsic goal orientation is stronger for the AIA group.

2) There is an increase in task value beliefs for both groups. Task value refers to the students' perception about the material of the course in terms of interest, importance, usefulness. Task value beliefs became stronger for the AIA group. Due to the widespread use of mobile devices among youngsters [28], students perceive programming for mobile devices to be more useful and important.
3) There is a small increase in the control of learning beliefs for both environments. Students believe that making more effort in programming with Scratch and AIA would lead to better results.
4) There is an increase in self-efficacy for both groups. Self-efficacy refers to students' confidence that they are able to perform well in the class.

During the second part of the study we will examine how motivation will progress towards the end of the course. Furthermore, gender influence on motivation will be examined. Student performance data (pre-test, assignments and final exam) in relation to motivational variables will be evaluated after course completion. Also, a larger number of participants, required for a quantitative study, will be used in a future work.

The current study contributes to understanding the transition in motivation to learn programming using Scratch and AIA. Based on these findings, educators could appropriately adjust their learning strategies to keep students interested and engaged in programming. This may keep students motivated towards computing and engineering disciplines as well.

REFERENCES

[1] Computing in the Core Newsroom, http:// www.computinginthecore.org/?/newsroom/representatives-brooks-and-polis-introduce-legislation-to-bolster-k-12, Accessed Nov. 11, 2013.

[2] The global race for STEM skills. *The Observatory on Borderless Higher Education Report.* Accessed November 11, 2013, from http://www.obhe.ac.uk/newsletters/borderless_report_january_2013/global_race_for_stem_skills.

[3] Digest of Education Statistics: 2010, http://nces.ed.gov/programs/digest/d10/, Accessed Nov. 11, 2013.

[4] M. Guzdial, "Programming environments for novices", In S. Fincher and M. Petre, (Eds.), *Computer science education research*, pp.127-154, Taylor & Francis, 2004.

[5] C. Kelleher and R. Pausch, "Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers", *ACM Computing Surveys*, Vol. 37(2), pp.83-137, 2005.

[6] S. Uludag, M. Karakus, and S.W. Turner, "Implementing IT0/CS0 with scratch, app inventor for android, and lego mindstorms", In *Proceedings of the 2011 conference on Information technology education*, pp.183–190. New York, 2011.

[7] M. Anderson and C. Gavan, "Engaging undergraduate programming students: experiences using lego mindstorms NXT", In *Proceedings of the 13th annual conference on Information technology education*, pp. 139–144, New York, 2012.

[8] C. Kelleher, R. Pausch, and S. Kiesler, "Storytelling Alice motivates middle school girls to learn computer programming", In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, San Jose, California, 2007.

[9] M. Kölling, "The Greenfoot Programming Environment", *Transaction on Computing Education*, Vol. 10 (4), 14:1-14:21, 2010.

[10] B. Moskal, D. Lurie, and S. Cooper, "Evaluating the effectiveness of a new instructional approach", In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, Norfolk, VA, March 03-07, 2004.

[11] K. Roy, "App inventor for android: report from a summer camp", In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pp. 283–288, New York, 2012.

[12] K. Brennan and M. Resnick, "Stories from the scratch community: connecting with ideas, interests, and people", In *Proceeding of the 44th ACM technical symposium on Computer science education*, pp. 463–464, New York, 2013.

[13] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The Scratch Programming Language and Environment", *Transaction on Computing Education*, Vol. 10(4), 16:1–16:15, 2010.

[14] O. Meerbaum-Salant, M. Armoni, and M. Ben-Ari, "Learning computer science concepts with scratch", In *Proceedings of the Sixth international workshop on Computing education research*, pp. 69–76. New York, 2010.

[15] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner E. Rosenbaum, J. Silver, B. Silverman, and Y. Kafai, "Scratch: programming for all", *Communication ACM*, Vol. 52(11), pp.60–67, 2009.

[16] U. Wolz, J. Maloney, and S.M. Pulimood, 2008. "'Scratch' your way to introductory *CS".* In *SIGCSE'08,* Portland, Oregon, USA, 2008.

[17] J. Gray, H. Abelson, D. Wolber, and M. Friend, "Teaching CS principles with app inventor". In *Proceedings of the 50th Annual Southeast Regional Conference*, pp. 405–406, New York, 2012.

[18] S. Grover and R. Pea, "Using a discourse-intensive pedagogy and android's app inventor for introducing computational concepts to middle school students", In *Proceeding of the 44th ACM technical symposium on Computer science education*, pp. 723–728, New York, 2013.

[19] R. Morelli, T. de Lanerolle, P. Lake, N. Limardo, E. Tamotsu, and C. Uche, "Can Android App Inventor Bring Computational Thinking to K-12?", In *Proc. 42nd ACM technical symposium on Computer science education (SIGCSE'11),* 2011.

[20] S. Papert, "Mindstorms: children, computers, and powerful ideas", NY, USA: Basic Books, Inc., 1980.

[21] J. R. Savery, "Overview of Problem-based Learning: Definitions and Distinctions", *Interdisciplinary Journal of Problem-based Learning*, Vol.(1)1, 2006.

[22] K.D. Simons and P.A. Ertmer, "Scaffolding disciplined inquiry in problem-based learning environments", *International Journal of Learning*, Vol.12, pp.297–305, 2006.

[23] C.E. Hmelo-Silver, "Problem-based learning: What and how do students learn?", *Educational Psychology Review*, Vol. 16, pp.235 – 266, 2004.

[24] L. Wijnia, S. M. M. Loyens, and E. Derous, (2011). "Investigating effects of problem-based versus lecture-based learning environments on student motivation" , *Contemporary Educational Psychology*, Vol. 36, pp. 101–113, 2011.

[25] P. R. Pintrich, D.A. Smith, T. Garcia, and W.J. McKeachie, "A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ). National Center for Research to Improve Postsecondary Teaching and Learning", Ann Arbor: University of Michigan, 1991.

[26] D. Wolber, "App inventor and real-world motivation". In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pp. 601–606, New York, 2011.

[27] H. Tsukamoto, H. Nagumo, Y. Takemura, and K. Matsumoto, "Analyzing the transition of learners' motivation to learn programming", In *38th Annual Frontiers in Education Conference Proceedings*, 2008.

[28] http://www.nielsen.com/us/en/newswire/2012/young-adults-and-teens-lead-growth-among-smartphone-owners.html, Accessed Jan 5, 2014.