



Using Gamification and Cyber Learning Environment to Improve Students' Learning in an Introductory Computer Programming Course: An Empirical Case Study

Mrs. Mourya Reddy Narasareddy Gari, North Dakota State University

I am Ph.D student at North Dakota State University. My research work is to see how different Learning strategies affect the student learning.

Dr. Gursimran Singh Walia, North Dakota State University

Gursimran S. Walia is an associate professor of Computer Science at North Dakota State University. His main research interests include empirical software engineering, software engineering education, human factors in software engineering, and software quality. He is a member of the IEEE Computer Society. Contact him at gursimran.walia@ndsu.edu

Mr. Alex Radermacher, North Dakota State University

An Empirical Study of Usage of Cyber Learning Environment to Improve Student's Learning and Engagement in an Introductory Computer Programming Course

1. Introduction:

Writing computer programs is at the core of Computer Science (CS) and Software Engineering (SE) courses. It is therefore imperative for CS and SE educators to teach students the basic programming concepts beginning with introductory programming courses and better prepare them for their jobs in software industry. Due to the increasing need of automation, CS and SE workers in the US are at great demand. However, researchers have indicated that there are several knowledge deficiencies [1] in the learning outcomes of the students who are graduating from Computer science and software engineering courses such that those students are unable to meet the expectations of the industry. This result can be traced back to lack of understanding of fundamental programming concepts leading to development of poorly developed software. Part of this may be a result of students and instructors lack access to vetted learning content on programming concepts leading to high dropout rates in introductory programming courses [8].

While there can be numerous factors that impact a student's decision to drop out of an introductory programming course, there is data to support that this problem is prevalent across institutions that report upwards of 40 percent dropout rates in introductory CS programming courses [19-20]. The most commonly reported reasons include lack of resource support, engagement and motivation [9]. To help students improve their understanding of programming concepts and become better programmers, we aim to provide them with a cyber-learning platform that incorporates an array of learning engagement strategies (e.g., collaboration, social networking, gamification).

To that end, we introduce and subsequently validate the usefulness of our cyber learning environment with helping students improve their fundamental understanding of introductory computer programming concepts. The environment presented in this paper is SEP-CyLE (Software Engineering and Programming – A Cyber Learning Environment) [2] that contains a repository of vetted learning objects (LOs) and tutorials and uses a combination of learning engagement strategies to get motivate students to be more involved in learning programming concepts contained in LOs. More details on SEP-CyLE and engagement strategies are provided in Section 3.

This paper reports the results from an empirical study conducted at North Dakota State University that investigated the impact different combinations of engagement strategies in SEP-CyLE had on the students' acquisition of programming concepts and engagement. Specifically, we evaluated SEP-CyLE with and without collaborative learning and evaluated its impact on undergraduate student's improvement in programming knowledge and skills.

The remainder of the paper is organized as follows. Section 2 provides some background and related work on gamification and team collaboration. Section 3 details the features of SEP-CyLE and explains both the instructor and student view of the tool. Section 4 describes the study design, the participating students, the study procedure, and the data collected during the

experiment. Section 5 describes the data analysis and results of the study. Threats to validity are presented in section 6. Finally, section 7 concludes the paper and presents ideas for future work.

2. Background and Related work:

In this section, we introduce learning engagement strategies used in SEP-CyLE. This section will briefly describe learning engagement strategies and provide evidence on benefits of using them in a cyber learning environment.

Gamification: Gamification is normally defined as the usage of game design elements in non-game contexts [3]. These gamification elements that are found and used in most of the researches include awarding points, leader board, and badges among other strategies [16]. The success of gamification in education is tied to its potential to connect with students by engagement, and its employment has been linked to student success [4-6]. Gamification has also been successfully applied to improve learning in health and education [13-16]. Our study is focused on reward points and leaderboard and evaluating their impact on student learning.

Collaborative learning: Collaborative interactions among individuals, has shown positive results across different levels of education, ranging from young children doing their school projects like craft work in teams to university students working on development projects [17],[21]. At the same time, it is essential to make a distinction between team performance and team effectiveness. This is because team performance represents the results of group's activities while paying little heed to how the group have accomplished the task. Team effectiveness, however, takes a more holistic point of view in considering not only whether the group performed effectively, but also how the group cooperated to accomplish the group task [18]. In our case study, we examine virtual teams (i.e., teams on SEP-CyLE do not have to work together directly) and as such their team score is based on collective completion of individual work.

Social interaction: Collaborative learning prompts further deeper level learning, critical thinking, shared understanding, and long term maintenance of the learned material [7], [22]. Despite the fact that various factors (e.g. nature of task, learning styles, group composition, etc.) have been distinguished as components which possibly impact the viability of collaborative learning, social interaction is most commonly referenced as key to effective collaboration [7, 12]. SEP-CyLE includes all of the above engagement strategies as detailed in the next section.

Digital Learning objects (LOs): The central to the design of SEP-CyLE are deployment of digital learning objects (prepared and vetted by experts) to promote personalized study and enhance learning. David Wiley describes LOs as small chunks of knowledge that are self-contained and re-usable [23]. SEP-CyLE contains an array of LOs spanning computer programming and software engineering concepts.

3. SEP-CyLE – Software Engineering and Programming Cyber Learning Environment:

This section provides an introduction of the SEP-CyLE, its main features and how it can be used in programming courses by instructors (or TA's) and their students.

3.1. Overview, current design and features of SEP-CyLE:

SEP-CyLE [2] is an online repository that contains vetted learning objects developed to assist instructors and broaden the participation of undergraduate students exposed to software programming concepts in undergraduate CS and SE courses. SEP-CyLE provides students with reading tutorials which comprised of a variety of software programming concepts and methodologies. Students can browse through LOs and tutorials, and then analyze their understanding of those concepts through quizzes. SEP-CyLE incorporates the following learning engagement strategies for improving student engagement:

Social features: These features allow a student to set up and customize their user profile, pose questions or discuss any topic in the discussion forum. For each of these activities the students gain virtual points, which they can keep track of.

Team collaboration features: These features allow student to view their team members. Each of the student's individual performance will be taken into consideration and their team score depends on aggregate fulfillment of individual work. For teams completing the task, based on their speed of completion, each team is awarded team virtual points, apart from their individual virtual points.

Learning Objects (LO): These are the vital components of SEP-CyLE and contain a bulk of the education material in the system. LOs are presented in various forms including text, audio, and video and are designed to serve digestive chunks of information that can be completed in fifteen minutes or less. Each LO contains quizzes in the form of a set of question and answers to allow students to test their understanding of the content. LOs are self-contained and help students to become knowledgeable in specific topics.

Gamification: Gamification within SEP-CyLE has been designed and implemented with the objective of improving student engagement and motivation through playful and context-centric activities. Virtual (reward) points and leaderboards are the key gamification elements employed in SEP-CyLE. Virtual points are earned by the student by completing LOs or engaging in other social activities on SEP-CyLE. Also, the leaderboard feature helps the student know where they are standing in the class and helps increase engagement by motivating student to go for higher positions on the leaderboard. Leaderboard information can be seen by both the instructor and student. Since students often report lack of interest and engagement while learning software testing, gamification can serve to both enhance student motivation and engagement while at the same time providing feedback on the student's level of competency.

3.2.SEP-CyLE: Instructor View:

The instructor view has been divided into 4 sections:

- Student Management (uploading the student roster, managing student teams)
- Course Communication Management (monitoring activity streams, course forums)
- Assignment Management (allocation of Learning Objects, tutorials)
- Miscellaneous (generation of student reports, enabling the features of the course)

Student management: Instructors can use SEP-CyLE at their campus by requesting access from the administrators of SEP-CyLE [2], who provide login credentials to the instructor and create a course for the instructor. Following that the instructor can upload their student roster, create unique passwords for the class/students (which students can change later on), assign students to virtual teams, and create a rubric for how virtual points are awarded based on student activities.

Course communication management: Instructors can create and monitor forums where students can hold conversations on different software programming and testing topics. Instructors are able to add new discussion topics or edit/delete the existing ones. Also, the instructor can view student discussion on course forums which may provide them with an additional idea of how students are understanding the concepts.

Assignment management: This aids the instructor in assigning the LOs and tutorials to the students. Instructors are able to set the deadline to finish these tasks, determine any minimum passing score for an LO's quiz, and adjust the number of attempts allowed for a quiz.

Miscellaneous: This assists an instructor in monitoring students' activities in SEP-CyLE. This includes information such as the number of times a student has visited a tutorial, the amount of time spent on each LO, the amount of time taken to solve a quiz of particular LO. It also allows instructors to view the number of virtual points gained by each and any of the students in the class, to view the class average scores for each LO, and other pieces of information associated with student performance or engagement. The course settings also allow the instructor to enable or disable different gamification features for the course.

3.3. SEP-CyLE: Student View:

Once the instructor creates an account for the students, students will be able to log in to SEP-CyLE using the passwords created for them by the instructor. Students can then create their user profile (adjusting things like their name and profile picture), change their password, browse through the learning objects and tutorials, take quizzes, communicate with other students using the discussion forums, and view their course posts. The dashboard of the student consists of the learning objects and tutorials. The dashboard gives the student what are all the learning objects and the tutorials that are available to them. They had practice quizzes (with feedback) and graded quizzes that they needed to pass with at least 80% correctness to earn virtual points.

Students also have the ability to view community posts, their own virtual points and course leaderboard, view their team members (if teams have been assigned by the instructor), and any recent course activity such as LOs that have been assigned by the instructor or activities by other students in the course. Examples of course activities include completing a quiz, making a post in the forms, or team completion for some common task.

4. Study Design

The study was designed to investigate the effect of SEP-CyLE (and associated features) on undergraduate students' acquisition of programming concepts, engagement, and course performance in an introductory computer programming course at North Dakota State University. The study was conducted across two different sections of an introductory computer programming course (CS1) taught by the same instructor. The first section (the experimental group) had all SEP-CyLE features (social networking, gamification, and team collaboration)

enabled whereas the second section (the control group) section had everything enabled except the team collaboration. That is, the control group section did not include student teams and team based virtual points. This was done to isolate the impact of collaboration (in SEP-CyLE) on student performance. The study used pre- and posttest instrument to quantify the impact of utilizing SEP-CyLE (with or without team collaboration) on student's acquisition of programming concepts and their course performance. The pretest was conducted at the very beginning of the semester, before the introduction of SEP-CyLE to students. Next, the students were asked to work individually and complete their assigned programming labs and other course work. At the end of the semester the students were tested again using the posttest instrument which used the same set of questions as in pretest. More details of the study are provided in the accompanying sections.

4.1. **Study goals:**

The study has two goals. The main goal of the study is to investigate the impact the SEP-CyLE had on the student's learning of software programming concepts which is stated in the form of the following research question:

Does the inclusion (or exclusion) of collaborative learning in SEP-CyLE impact students' acquisition of programming concepts and their performance in an introductory programming course?

The above question isolated the impact the team collaboration (by comparing performance of students in section with and without team collaboration) had on the student engagement and their course performance.

4.2. **Independent and dependent variables:**

The independent variables of the study are:

- i. *Team collaboration points:* These are the virtual points that are earned by the students on completion of a group task.

The dependent variables that will be measured are:

- i. *Virtual Points:* These are the reward points that are earned by the students for activities within SEP-CyLE.
- ii. *Total grade:* This is the course grade that is received by the students, which includes grades on exams and programming exercises.
- iii. *Number of LO's the students attempted* measured the number of learning objects that each student attempted
- iv. *Number of LO's completed* measured the number of learning objects that a student passed with at least 80% of questions correctly answered

4.3. **Participating Subjects:**

The case study was conducted across two sections of introductory programming course at NDSU. One section had 46 students and the other had 47 students which totals to 93 students.

35 out of 93 students chose to participate in the study (i.e., these students completed both of pre/post tests and provided consent to participate).

4.4. Study Procedure:

Pretest: Prior to beginning of the study, a pretest was conducted to establish a baseline for the students’ understanding of computer programming concepts. The pre-test questions are available in Appendix A. The test had 21 questions that evaluate different programming concepts (e.g., arrays, pointers, software testing, and memory management). A mapping of different questions to programming concepts is detailed in Section 5. The questions in the pre/post-test were graded by awarding one point for each correctly answered question.

Introduction of SEP-CyLE: All the subjects are trained on how to use the SEP-CyLE tool by introducing the features of the tool: viewing tutorials, taking quizzes, and posting and viewing of the discussion threads. The control group section had access to all of the same learning objects and other material, but were not assigned to virtual teams. The experiment group section was introduced to the team collaboration aspects of SEP-CyLE. Students earn individual virtual points for successful completion of each LO and also earned team-based virtual points if all team members were able to successfully complete an LO. Table 1 presents the point allocation settings for experiment and control group:

Table 1: Point Allocation Settings for both the sections

	<i>Number of points</i>	<i>Control Group</i>	<i>Experiment group</i>
<i>Quiz Complete</i>	3	✓	✓
<i>Profile Picture Upload</i>	2	✓	✓
<i>Course Thread Post</i>	1	✓	✓
<i>Team Complete</i>	1	*	✓
<i>First Team Complete</i>	3	*	✓
<i>Second Team Complete</i>	2	*	✓
<i>Third Team Complete</i>	2	*	✓

Posttest: At the end of the semester, the students were again evaluated on their understanding of programming concepts using a posttest instrument. This posttest has the same set of questions that are present in pretest. This posttest was given in order to evaluate any improvement in students’ knowledge or proficiency related to software programming concepts, methodologies, and tools they have gained through exposure to SEP-CyLE. We also interviewed the course instructor to gain insights into the usability of SEP-CyLE and to improve future usage.

5. Data Capture:

We collected data for the number of virtual points earned by each student at the end of the semester and logs (in terms of time spent on SEP-CyLE) for different types of activities in the SEP-CyLE system. We also collected data regarding the number of LOs attempted and completed and the course grade for all students.

We analyzed student activity on SEP-CyLE (e.g., time spent, LOs attempted and completed, virtual points earned) to find correlations between the usefulness of SEP-CyLE features and

student course performance (i.e., course grades). Additionally, we analyzed their pre- and posttest scores to evaluate improvement in their understanding of computer programming concepts.

To understand their improvement in understanding of specific programming constructs, the pre/posttest questions were mapped to 8 specific programming concepts (where multiple questions corresponding to different programming concepts as shown below). The mapping of questions to specific concepts are as follows:

Table 2: Categories of questions

Categories	Question Numbers
Memory Management	1, 2, 12 and 15
Basic Programming Concepts	3, 4, 8, 9 and 22
Arrays	5, 11
Basic Operators	6 and 21
Methods	7
Pointers	10
OOPs Concepts	13, 14 and 16
Software Testing Concepts	17, 18, 19 and 20

Pre- and post-test analyses were separately performed on each of these categories for control group and experiment group.

6. Results and analysis:

The results have been organized around the research questions introduced in section 3. Section 6.1 compares the pre- and post-test results to evaluate the improvement in students' programming conceptual knowledge before and after using the SEP-CyLE. Additionally, this section presents the impacts of using SEP-CyLE on student course grade and the effects of team collaboration on course performance.

6.1. Pretest vs Posttest Results:

To evaluate the impact of SEP-CyLE on the student's knowledge acquisition of the programming concepts and their proficiency of tools and techniques, a comparison of pre- and posttest was performed. Students were given approximately 20 minutes to complete each of the tests. The average score of pretest was compared against the average score of posttest for control group and experiment group. Figure 1 showed higher increase in the post-test score for the control group (that did not have the team collaboration) as compared to the experiment group (that included team collaboration). The result from paired t-test showed that the increase was statistically significant (at $p < 0.001$) for both groups.

We had expected that ability to collaborate and earn extra virtual points when working with peers would motivate students to perform better, however the results seem to indicate otherwise. Based on the feedback from instructors, it was found that teams were formed randomly (size of 3 students) which limited the engagement and that could have resulted with less team-based collaboration within SEP-CyLE. In future work, we plan to evaluate the impact of team collaboration by forming teams based around those used for group assignments to establish if

team collaboration on cyber learning environment (SEP-CyLE) can have more pronounced impact on students' learning.

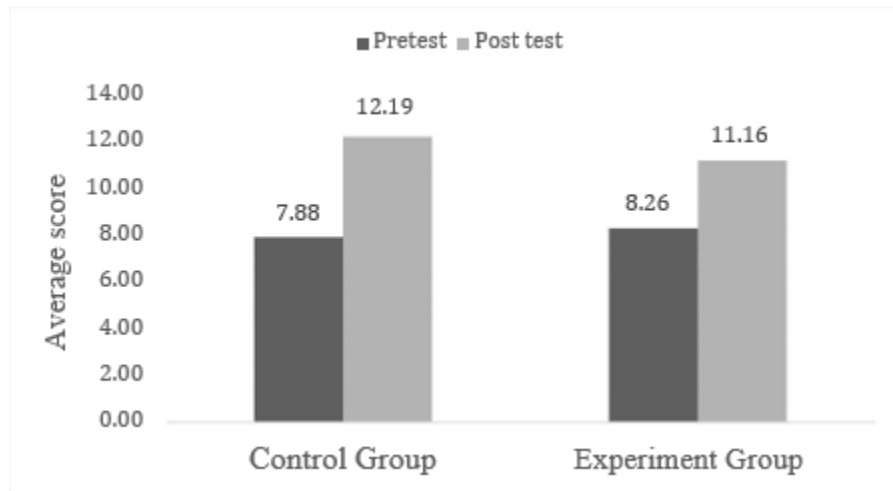


Figure 1: Mean Pre/Post Test Value of Control Group & Experiment Group

To gain further insights into students' acquisition of programming concepts, we compared their pre- vs. post-tests for each of 8 categories listed in Table 2 (e.g., memory management, testing, methods, and arrays). The resulting comparison for each of 8 programming concepts is shown in Figure 2. To perform this analysis, we combined the subjects in both groups (experiment and control groups) to enable a larger data point comparison. That is, for each category, pre-test scores for all 35 subjects were compared against the post-test scores for all 35 subjects (because all subjects used SEP-CyLE).

We also performed paired t-test to evaluate whether the improvement in post-tests was statistically significant for all detailed programming concepts. Based on the results, other than Methods (p-value of 0.023); all other categories showed statistically significant improvement (p-value <0.001) when comparing pre- vs. post-test scores for all 35 students. Based on the instructor feedback, once SEP-CyLE is populated with more learning objects (especially for concepts where students are still lacking during the post-test); their knowledge of programming concepts and usage of SEP-CyLE is expected to increase further.

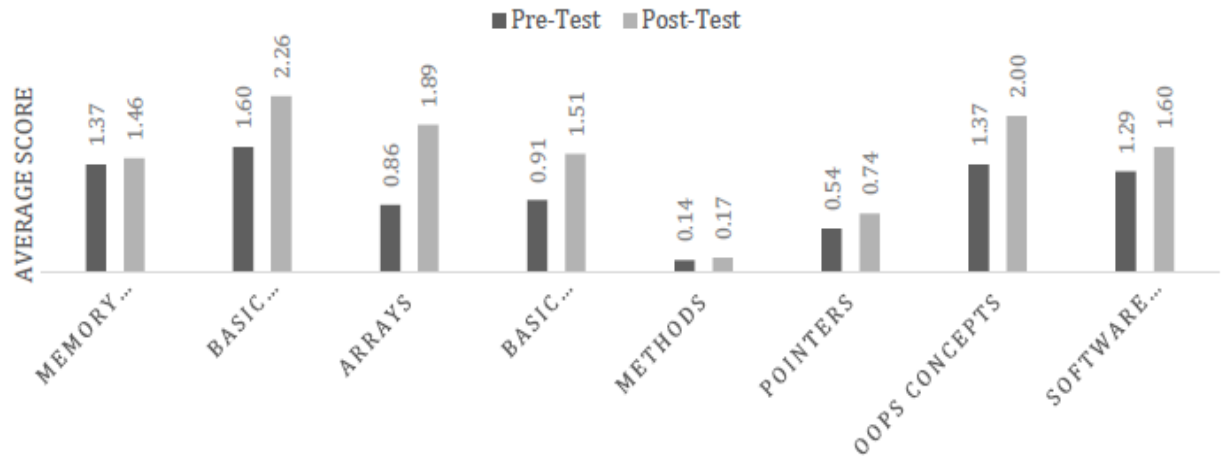


Figure 2: Mean Pre/Post Test Value for all programming concept categories

6.2. Impact of SEP-CyLE activity vs. Course Performance:

An important aspect of this study was to analyze the impact different versions of SEP-CyLE had on student engagement. Student engagement was measured in terms of the amount of time spent on SEP-CyLE, number of LOs attempted (or completed), and virtual points earned. We compared the engagement measures in experiment group (SEP-CyLE with everything enabled) vs. control group (SEP-CyLE without team collaboration). Based on the results, the students spent significantly more time on SEP-CyLE when collaborative learning was disabled (i.e., control group) when compared to the experiment group. This was contrary to the expectations. Interestingly, more time spent on SEP-CyLE did not translated to additional virtual points (i.e., both groups fared equally well on virtual point allocations).

We also conducted correlation tests (shown in Table 3) to analyze the relationship between SEP-CyLE metrics (time spent on SEP-CyLE, number of virtual point earned, number of LOs completed, and number of LOs attempted) against their course performance (i.e., their end of semester grades).

Table 3: Correlation Results of the Experiment group

	Control Group		Experiment Group	
	p-value	Pearson Correlation	p-value	Pearson Correlation
# of LO's attempted vs. Course performance	.067	.469	.949	.016
# of LO's Completed vs. Course performance	.071	.463	.827	.055
Virtual points earned vs. Course performance	.157	.371	.164	.333
Time spent on SEP-CyLE vs. Course performance	.825	.060	.129	.372

The correlation results showed a positive and significant relationship between the number of LOs attempted (p-value of 0.067) and the number of LO's completed (p-value of 0.071) and their course grade for the control group as highlighted (with shaded cells) in Table 3. Additionally, the correlation between the virtual points earned by students on SEP-CyLE was positively (though non-significant) correlated with their course performance for both control and experiment groups. Based on this result the SEP-CyLE LO's had a positive impact on students' performance in an introductory computer programming course.

7. Discussion:

The results from the study indicate a general effectiveness when using SEP-CyLE with the gamification features to teach programming concepts, methodologies and testing in introductory computer programming courses. The students' understanding of the basic programming and software testing concepts, and their proficiency with the other areas such as arrays, methods, pointers, etc., all showed improvement. An interesting result was that the introduction of team collaboration points in conjunction with the gamification features like leaderboards in the course did not result in a significant difference in the performance of the students exposed to the team collaboration features. Although unexpected, this could have been in part due to the fact that virtual teams on SEP-CyLE were randomly selected and were not same as teams that worked together on course assignment and projects which may have impacted their motivation. Additionally, the collaborative tasks in SEP-CyLE (i.e., team virtual points) did not require interaction among team members as long as each member of team completed an LO individually. Collaborative learning generally involves significant teamwork (sharing knowledge, resolving differences and building consensus) which encourages engagement.

The positive correlation between the performance of the students and the virtual points they have earned and also the positive correlation between the number of LOs attempted by the students and their grades, indicates that the SEP-CyLE did have an impact on the performance of the students in a positive way.

8. Threats to validity:

This section attempts to address some of the threats to validity that may have affected the results of the research. A major threat is small size of subjects participating in this study which limited the data analysis. Second, the students may not have taken the posttest seriously because it was voluntary (i.e., did not had an impact on grades) and they might not have been motivated/inspired to do well on it.

9. Conclusion:

In this paper we described our approach of investigating team collaboration in SEP-CyLE in conjunction with gamification elements. We discussed the current design and the results of the case study in order to determine the impact of SEP-CyLE with team collaboration on the students learning. Our results have indicated that while the use of SEP-CyLE with gamification elements can positively impact students' performance and engagement in an introductory programming course, the inclusion of team collaboration did not have any effect on the student learning outcomes and was not associated with increased engagement with SEP-CyLE.

Motivated by these results, we plan to see how the SEP-CyLE cyber-learning environment would impact the performance of students when team formation is based around groups that are assigned to work on projects outside of SEP-CyLE. Also, we are working with course instructors to develop a larger set of LOs that would help overcome some of the knowledge deficiencies of students and would enable increased usage of SEP-CyLE in introductory programming courses.

10. Acknowledgements:

This work is supported in part by the National Science Foundation under grants DUE-1225742 and DUE-1525112. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

11. References:

- [1] R. Alex & G. Walia & K. Dean. (2015). Missed expectations: Where CS students fall short in the software industry. *CrossTalk*. 28. 4-8.
- [2] R. Chang-lau and P. J. Clarke. Software engineering and programming cyberlearning environment (SEP-CyLE), 2018. <https://stem-cyle.cis.fiu.edu/instances> (retrieved Jan. 2018).
- [3] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, (2011). "From game design elements to gamefulness: defining gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pp 9-15.
- [4] G. D. Kuh, "Assessing what really matters to student learning inside the national survey of student engagement," *Change: Mag. Higher Learn.*, vol. 33, no. 3, pp. 10–17, 2010.
- [5] K. Krause and H. Coates, "Students' engagement in first-year university," *Assessment Eval. Higher Edu.*, vol. 33, no. 5, pp. 493–505, 2008.
- [6] OECD, PISA 2009 Results: Learning to Learn—Student Engagement, Strategies and Practices, vol. 3. Paris, France: OECD Publishing, 2010.
- [7] K. Karel & K. Paul & J. Wim. (2003). Identifying the Pitfalls for Social Interaction in Computer-Supported Collaborative Learning Environments: A Review of the Research. *Computers in Human Behavior*. 19. 335-353. 10.1016/S0747-5632(02)00057-2.
- [8] S. D. D. Dixon, R. Khaled, and L. Nacke, "Gamification: Toward a definition," CHI 2011 gamification workshop, 2011.
- [9] D. Sebastian, et al. "From game design elements to gamefulness: defining gamification." *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. ACM, 2011.
- [10] M. Niitsoo, M. Paales, M. Pedaste, L. Siiman, & E. Tõnisson, (2014). Predictors of Informatics Students' Progress and Graduation in University Studies. e European *Procedia Social and Behavioral Sciences* (x). Elsevier (avaldamisel).

- [11] P. Kinnunen, and L. Malmi “Why Students Drop Out CS1 Course?” *Proceedings of the Second International Workshop on Computing Education Research*: 97–108, 2006.
- [12] D.R. Garrison “Quality and theory in distance education: theoretical consideration” D. Keegan (Ed.), *Theoretical principles of distance education*, Routledge, New York (1993).
- [13] C.A. Bodnar, D. Anastasio, J.A. Enszer, & D. D. Burkey, (2016). Engineers at Play: Games as Teaching Tools for Undergraduate Engineering Students. *Journal of Engineering Education*.
- [14] O. Pedreira, F. García, N. Brisaboa, & M. Piattini (2015). Gamification in software engineering—A systematic mapping. *Information and Software Technology*, 57, 157-168.
- [15] H. Arksey & L. O'Malley (2005). Scoping studies: towards a methodological framework. *International journal of social research methodology*, 8(1), 19-32.
- [16] J. Hamari, J. Koivisto, and H. Sarsa, “Does Gamification Work?—A Literature Review of Empirical Studies on Gamification” in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, 2014, pp. 3025–3034.
- [17] D. Pierre & J.Sanna & Fischer, Frank & Balacheff, N & Ludvigsen, Sten & De Jong, T & Lazonder, A & Barnes, Sally. (2018). The evolution of research on computer-supported collaborative learning: from design to orchestration.
- [18] E. Salas., Sims, D. E., & Burke, C. S. (2005). Is there “big five” in teamwork? *Small Group Research*, 36(5), 555–599.
- [19] B. Theresa and J.Mason. “Why the high attrition rate for computer science students: some thoughts and observations.” *SIGCSE Bulletin* 37 (2005): 103-106.
- [20] J.Bennedsen, M.E. Caspersen, Failure rates in introductory programming, *ACM SIGCSE Bulletin*, v.39 n.2, June 2007 [doi>10.1145/1272848.1272879].
- [21] DW. Johnson, RT. Johnson, K Smith. The state of cooperative learning in postsecondary and professional settings. *Educ Psychol Rev.* 2007;19:15–29.
- [22] D.R. Garrison, T. Anderson, W. Archer Critical thinking and computer conferencing: a model and tool to access cognitive presence. *American Journal of Distance Education*, 15 (1) (2001), pp. 7-23.
- [23] D. Wiley, “Learning Objects, Content Management and E-Learning,” *Content Management for E-Learning*, N. Ferran and J. Minguillo ´n, eds., chapter 3, Springer, 2011.