Teaching Problem Solving Skills using an Educational Game in a Computer Programming Course

Roy MATHEW, Sohail Iqbal MALIK, Ragad Moufaq TAWAFAK

Department of Information Technology, Buraimi University College, Al-Buraimi, Oman
e-mail: roy@buc.edu.om, sohail@buc.edu.om, raghad@buc.edu.om

Abstract. Problem solving skills are considered an important component in learning to program in an introductory programming (IP) course for novices. This study introduced a PROSOLVE game to enhance problem solving skills of novice programmers in the introductory programming course. The game is based on pseudo-code technique. A survey was employed to collect students' feedback and semi-structured interviews were organized to collect instructors' opinion about the game. The results show that the game helped most of the students in understanding the programming concepts, structures and problem solving strategies. Moreover, the game supports students' cognitive engagement, gains, and affective engagement in the IP course. Instructors appreciated the game and considered it as an additional supporting teaching tool in the IP course. Moreover, they considered the game as good alternative of traditional pen and paper learning approach in attracting students' interest in the programming domain.

Keywords: learning to program, problem solving skills, computer programming, pseudo-code technique, educational game.

1. Introduction

Programming is one of the demanding fields in the computer science discipline and 'employment of computer programmers is projected to grow 8 percent from 2012 to 2022' (Bureau of Labor Statistics, 2015). In 2012, Microsoft published a report showing that American colleges are not producing enough graduates in computer science to meet expected demand up to 2020 (Microsoft, 2012). On the other hand, coping with high failure and dropout rates in the IP course is another issue for computer science disciplines (Guzdial & Soloway, 2002; Lahtinen *et al.*, 2005; Sykes, 2007; Yadin, A. 2011; Watson & Li, 2014; Zingaro, 2015). Most of the computer science and related disciplines offer IP course at their first level and if students fail or drop out of this course, they are unlikely to enroll in a follow-on course (Wiedenbeck *et al.*, 2004). One of the

possible reasons is because the traditional approach used in teaching the IP course does not provide students all the required skills. Moreover, most of the introductory programming courses focus more on syntax and semantics of the programming language during the course and discuss problem solving strategies in the first few lectures only (Malik, 2018).

This study introduced the PROSOLVE game to enhance the problem solving skills of novice programmers in the introductory programming (IP) course. The game is based on pseudo-code technique. The game covers most of the teaching topics of the IP course such as algorithm and pseudo code, basic elements of C++, input/output, selection control structure, repetition control structure, functions and arrays. These topics represent as seven levels of the game as shown below in Fig. 1 as a menu bar. Consequently, students practice the problem solving skills throughout the course. We propose two research questions to guide the study and to determine the impact of the PROSOLVE game on novices.

The two research questions being addressed in this research study are as follows:

RQ1: *What are the perceptions of students regarding the PROSOLVE game in the introductory programming course?*

RQ2: *What are the perceptions of instructors regarding the PROSOLVE game in the introductory programming course?*

This paper is divided into a number of sections. It starts with a literature review, followed by a brief introduction to the PROSOLVE game. The methodology adopted for this study is then described, and research results are reported and discussed. The paper concludes with a summary of the outcomes.

2. Literature Review

Students of the 21st century are considered as digital natives and computer is part of their everyday life. Students start using computer even before they first attend formal education. Moreover, students are significantly accustomed to many of computer's functionalities because they used to play computer games from a very early age. Consequently, substantial number of instructors suggests incorporating educational games in the teaching and learning process of the programming courses to motivate students, and pique their curiosity in the programming domain (Malliarakis *et al.*, 2014; Akcaoglu, 2016).

On the other side, different studies reported high failure rates and lack of students' interest in the programming courses (Lahtinen *et al.* 2005; Sykes 2007; Guzdial and Soloway 2002; Kinnunen and Malmi 2006; Malik and Coldwell-Neilson 2017a). One of the challenges for IP students is to acquire a different set of skills (problem solving, program design, syntax and semantics) at the same time (Shuhidan 2012). The IP courses pay more attention on syntax and semantics of the programming language (Iqbal and Harsh, 2013). Moreover, some textbooks used in the IP courses also dis-

cussed problem solving strategies in the early chapters only (De Raadt, 2008). The ADRI (Approach, Deployment, Result, improvement) approach was proposed in the IP course to pay equal attention on problem solving strategies and programming knowledge (syntax and semantics) (Sohail, 2019). Two problem solving strategies (flowchart and pseudo-code) were used in the Approach stage and programming knowledge was discussed in the Deployment and Improvement stages (Malik and Coldwell-Neilson, 2017b; Malik, 2016a). The ADRI editor was also prepared on the bases of the ADRI approach to reinforce problem solving strategies and programming knowledge in the course (Malik, 2016b).

Gunter *et al.* (2008) proposed to incorporate educational games or series games in the teaching and learning of the programming courses to address the problems faced by novice programmers. These games help students to achieve specific goals, view their success immediately which enhances their self-confidence and decision making skills (Malliarakis *et al.*, 2014). Different frameworks have been proposed and developed for educational games in some research studies (Becker, 2010; Yusoff *et al.*, 2009; Zualkernan, 2006). Becker (2010) proposed the framework which suggested cognitive and emotional axes to investigate the goals of educational games. Cognitive axis covers mental competencies and educational goals set in this axis should meet all the six categories of the Bloom's taxonomy in educational games. The educational goals set in the emotional axis should cover emotional areas such as attitude and motivation.

Some educational games have been specifically developed for computer programming courses. Malliarakis *et al.* (2014) sorted these educational games into two categories. The first category of games focus on teaching specific computer programming units and the second category of games cover multiple educational goals or teaching units. Educational games focus on teaching specific programming unit include Catacombs, Saving Princess Sera (Barnes *et al.*, 2007; Barnes *et al.*, 2008), EleMental: The Recurrence (Chaffin *et al.*, 2009), Wu's Castle (Eagle & Barnes, 2009), Robozzle (Li & Watson, 2011), LightBot (Piteira & Haddad, 2011), and TALENT (Maragos & Grigoriadou, 2011). On the other side, educational games focus on teaching multiple units of learning include Robocode (O' Kelly & Gibson, 2006), M.U.P.P.E.T.S. (Phelps et. al, 2003), Prog & Play (Muratet *et al.*, 2010), PlayLogo3D (Paliokas *et al.*, 2011), and Gidget (Lee & Ko, 2011). All these games addressed both the cognitive and emotional axes during their development (Malliarakis *et al.*, 2014).

Battistella *et al.* (2017) introduced three educational games in data structure course for teaching Quicksort and Heapsort. They concluded that such educational games can contribute positively to the learning outcomes and increase the students' motivation in the course. Moreover, these games promote collaboration or competition among students and making learning more fun.

Theodoraki and xinogalos (2014) introduced arcade games as a means for learning programming in an undergraduate programming course. The result of students' survey shows that arcade games enhance students' motivation for learning programming, support students in comprehending complex programming concepts and engage students in carrying out programming activities.

Malik *et al.* (2017c) introduced a web based application (PROBSOL) in the introductory programming (IP) course to develop problem solving skills. The application is based on pseudo-code technique. They also introduced a mobile app of the application to facilitate students. The application helps students to focus on problem solving skills throughout the course. Results show that students appreciated both mode of learning (E-learning and M-learning) and attrition rate was reduced. They also suggested that games can play important role in teaching programming courses and can attract students' attention in the programming domain. Moreover, Sohail *et al.* (2019) concluded that students preferred M-learning more than the E-learning in the IP course.

3. Introduction to the PROSOLVE Game

The PROSOLVE is a web-based educational game. It was developed to enhance the problem solving skills of novice programmers in the IP course. It is based on pseudo-code technique. It covers all the teaching topics of the IP course such as algorithm and pseudo code, basic elements of C++, input/output, selection control structure, repetition control structure, functions and arrays. Fig. 1 shows the interface of the game.

The game consists of two list boxes (Random Steps and Solution) as shown in Fig. 1. The Random Steps list box shows all the Pseudo-code steps for a given problem statement in a random manner. A user re-arranges the given random steps of pseudo-code algorithm into its correct steps of sequential order by moving the steps, up or down using the 'Arrow Buttons' at right side of the list box. Number of clicks is recorded while moving the Arrow Buttons up or down. It displays a congratulatory
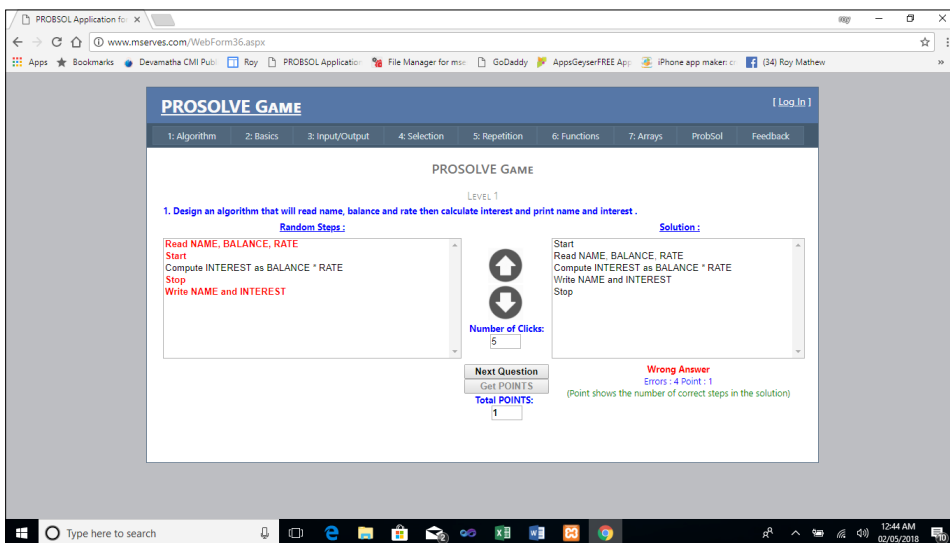

Fig. 1. List boxes of PROSOLVE game.

message, if proper sequence is already achieved after the last click for its correct solution.

When the users' solution is ready for the given problem statement, click at the command push button, 'Get POINTS'. It shows the correct answer along with the earned Total POINTS. It also displays the number of errors and points collected for correct sequential steps to the given question. Click at the button for 'Next Question' or navigate through the required menu option to get the next problem statement to solve it by providing its correct sequence of algorithm. Point shows the number of correct steps in the solution. Repeat these steps for all the questions specified in this current level and then the game goes to the next level. Questions related to another topic in the IP course are displayed in the next level.

There are 7 Levels in this game. The player can collect one point for each correct line of the answer. The player can win a Bronze Medal, if total collected points are in between 49 to 100. Win a Silver Medal, if total collected points are in between 99 to 200 and win a Gold Medal, if total collected points are 200 or more. The player will not get any medal if total points collected are less than 50. The player can get a Cup too when completes the game. But the type of the cup depends on the total points obtained. Different types of the cups are Black, Bronze, Silver and Gold. Black cup if total points collected are less than 50 and Gold cup if it is above 200 points.

4. Research Methodology and Design

The first research question was probed by conducting a survey with students after introducing the PROSOLVE game in the IP course. The second research question was investigated by conducting interviews with instructors of the IP course.

The population of the survey consists of students from the IP course at Buraimi University College, Oman. Fifty-one students participated in the survey including 41 female (80.4%) and 10 male (19.6%) students. The college offers academic programs in morning and evening shifts. Female students are allowed in both shifts and male students can register only in the evening shift. This process helps the college to attract more female students.

The semi-structured interview was performed with two instructors of the introductory programming course. An ethical approval was obtained from the college before collecting the data. Data collection was anonymous and student participation was optional and voluntary in the survey. Interviews data were reported without showing the instructors' identity.

4.1. *Preparation of the Survey*

The main objective of the survey is to determine the impact of PROSOLVE game on novice programmers in the Introductory Programming course. The survey consists of 27 closed-ended questions. The survey has three parts.

The first part of the survey includes demographic questions. It includes major, degree, age, gender, and prior programming experience of the students.

The second part of the survey consists of questions related to programming concepts and course content. This part includes 12 questions. A five point Likert scale (1 = strongly disagree to 5 = strongly agree) is used.

The third part of the survey focuses on questions which determine the acceptability of PROSOLVE game in the IP course. A five point Likert scale (1 = strongly disagree to 5 = strongly agree) is used. This part of the survey includes 10 questions.

5. Results

This section describes the results of this study and probes Research Questions 1 and 2. The first research question is as follows:

RQ1: What are the perceptions of students regarding the PROSOLVE game in the introductory programming course?

5.1. *Analysis of Survey*

A survey was conducted with the IP students after introducing the PROSOLVE game in the teaching and learning process of the course. RQ1 was explored by analyzing the responses of students in the survey.

The student survey consists of the following parts:

1. Demographic Details.
2. Programming Concepts.
3. PROSOLVE Game: Usability.

5.1.1. *Demographic Details*

The demographic details of the respondents are as given below:

- 37.3 % of respondents are enrolled in the Information Systems major, compared to 41.2 % from Computer Science and 21.6 % from Software Engineering.
- Most of the respondents (52.9 %) are pursuing their studies in Bachelor degree level as compared to (3.9 %) respondents in Advanced Diploma level and the remaining (43.1 %) in Diploma level.
- 94.1 % respondents are between 18 to 28 years old, compared to 2 % under 18 years of age, another 2 % of respondents were between age 29 to 44 and yet another 2 % are 45 years old or above.
- The majority of the respondents (80.4 %) were female as compared to 19.6 % male.
- Most of the respondents (70.6 %) do not have any previous programming experience.

5.1.2. *Programming Concepts*

The respondents' perception in understanding programming concepts after introducing the game is shown in Table 1. The means and frequencies of the responses are included in this table.

The results show that respondents perceive the three most significant areas are with 'Understand the programming concepts' (4.16), 'Learn the problem solving skill (Pseudo code)' (3.86), and 'Focus only on problem solving skills' (3.86). The respondents also found 'Design a program to solve a certain task' (3.67), 'Learn the course contents' (3.63), and 'Understand programming structures' (3.57) significant areas after introducing the game in the IP course.

Table 1

Programming Concepts

PROSOLVE Game helped me to:	Mean	Strongly Agree (% / N)	Agree (% / N)	Neutral (% / N)	Disagree (% / N)	Strongly Disagree (% / N)	Non Response (% / N)	Total (% / N)
Understand the programming concepts	4.16	25.5 % 13	64.7 % 33	9.8 % 5	0 % 0	0 % 0	0 % 0	100 % 51
Learn the problem solving skill (Pseudo code)	3.96	15.7 % 8	64.7 % 33	19.6 % 10	0 % 0	0 % 0	0 % 0	100 % 51
Focus only on problem solving skills	3.86	15.7 % 8	56.9 % 29	25.5 % 13	1.9 % 1	0 % 0	0 % 0	100 % 51
Understand programming structures	3.57	11.8 % 6	41.2 % 21	39.2 % 20	7.8 % 4	0 % 0	0 % 0	100 % 51
Learn the course contents	3.63	17.6 % 9	52.9 % 27	13.7 % 7	7.8 % 4	5.9 % 3	0 % 0	100 % 51
Design a program to solve a certain task	3.67	19.6 % 10	43.1 % 22	25.5 % 13	9.8 % 5	0 % 0	0 % 0	100 % 51

Table 2

Teaching topics

PROSOLVE Game helped me to learn the programming concepts of:	Mean	Strongly Agree (% / N)	Agree (% / N)	Neutral (% / N)	Disagree (% / N)	Strongly Disagree (% / N)	Non Response (% / N)	Total (% / N)
Designing an Algorithm	4.10	33.3 % 17	51.0 % 26	11.8 % 6	0 % 0	0 % 2	0 % 0	100 % 51
Input / Output	3.92	27.5 % 14	41.2 % 21	27.5 % 14	3.9 % 2	0 % 0	0 % 0	100 % 51
Selection Structure (if statement)	3.90	23.5 % 12	45.1 % 23	29.4 % 15	2.0 % 1	0 % 0	0 % 0	100 % 51
Repetition Structures (loops)	3.61	13.7 % 7	47.1 % 24	29.4 % 15	7.8 % 4	0 % 0	0 % 0	100 % 51
Functions	3.77	23.5 % 12	45.1 % 23	21.6 % 11	5.9 % 3	2.20 % 1	0 % 0	100 % 51
Arrays	3.75	19.6 % 10	49.0 % 25	21.6 % 11	7.8 % 4	0 % 0	0 % 0	100 % 51

The second part of this section covers questions related to the teaching topics of the IP course. Table 2 shows the means and frequencies of the responses for this section.

Respondents perceive 'Designing an Algorithm' (4.10), 'Input / Output' (3.92), and 'Selection Structure (if statement)' (3.90) as the most three significant learning topics. The programming concepts 'Functions' (3.77), 'Arrays' (3.75), and 'Repetition Structures (loops)' (3.61) were also perceived significant by respondents after introducing the game in the IP course.

5.1.3. *PROSOLVE Game: Usability*

The respondents' perception in motivating and engaging them in the IP course after introducing the game is shown in Table 3. The means and frequencies of the responses are included in this table.

Respondents perceive 'Motivates me to learn programming' (4.08), 'Engages me as an active learner in programming activities' (3.88), and 'Is effective to learn programming' (3.87) as the most three beneficial areas after introducing the game in the IP course. The respondents also found the PROSOLVE game 'Helps me to prepare for programming exams' (3.82), 'helps me to develop confidence in the subject area' (3.77), 'develops skills that apply to my academic career and /or professional life' (3.75), 'Is used at home after classroom activities' (3.73), 'helps me in the course activity in ways that enhanced my learning' (3.73), and 'Should also be prepared for advanced courses of programming' (3.63).

Table 3

Usability of PROSOLVE Game

PROSOLVE Game	Mean	Strongly Agree (% / N)	Agree (% / N)	Neutral (% / N)	Disagree (% / N)	Strongly Disagree (% / N)	Non Response (% / N)	Total (% / N)
Motivates me to learn programming	4.08	27.5 % 14	56.9 % 29	13.7 % 7	0 % 0	2.0 % 1	0 % 0	100 % 51
Engages me as an active learner in programming activities	3.88	17.6 % 9	58.8 % 30	17.6 % 9	5.9 % 3	0 % 0	0 % 0	100 % 51
Is effective to learn programming	3.87	17.6 % 9	52.9 % 27	29.4 % 15	0 % 0	0 % 0	0 % 0	100 % 51
Helps me to prepare for programming exams	3.82	19.6 % 10	51.0 % 26	21.6 % 11	7.8 % 4	0 % 0	0 % 0	100 % 51
Is used at home after classroom activities	3.73	23.5 % 12	43.1 % 22	21.6 % 11	5.9 % 3	5.9 % 3	0 % 0	100 % 51
Should also be prepared for advanced courses of programming	3.63	9.8 % 5	56.9 % 29	23.5 % 12	5.9 % 3	3.9 % 2	0 % 0	100 % 51
Helps me in the course activity in ways that enhanced my learning	3.73	9.8 % 5	58.8 % 30	27.5 % 14	2.0 % 1	2.0 % 1	0 % 0	100 % 51
Develops skills that apply to my academic career and /or professional life	3.75	15.7 % 8	54.9 % 28	23.5 % 12	0 % 0	5.9 % 3	0 % 0	100 % 51
Helps me to develop confidence in the subject area	3.77	7.8 % 4	66.7 % 34	21.6 % 11	2.0 % 1	2.0 % 1	0 % 0	100 % 51

5.2. *Semi-Structured Interviews of Introductory Programming Instructors*

This section explores the second research question which is:

RQ2: What are the perceptions of instructors regarding the PROSOLVE game in the introductory programming course?

Semi-structured interviews were conducted with the IP instructors to seek their perceptions regarding the PROSOLVE game in the introductory programming course. The interviews were conducted with two IP instructors. The main objectives of the interviews were to probe instructors' perceptions of:

1. Students' experience with the PROSOLVE game in the IP course.
2. The impact of the PROSOLVE game on students learning in the IP course and to explore.
3. Instructors' views on the strengths and weaknesses of the PROSOLVE game and to suggest.
4. Any further enhancements to the PROSOLVE game in the context of the introductory programming course.

5.2.1. *First Impression about the PROSOLVE game*

Both participants agreed that the overall impression about the game is very good. The game is easy to use and helps students to understand the basic concepts of the programming domain. One instructor liked the interactive type of approach adapted in the game which is useful for students to understand the concepts of programming. Another instructor pointed out that the questions asked to solve during this game is really targeting various topics of programming.

Both participants were satisfied to use this game in the IP course because it helps them to teach programming in a better way and they witnessed an easy learning from students' side. Fun elements in this application making students coming more attracted towards different topics of programming, which otherwise will be very difficult to achieve.

5.2.2. *Impact of the PROSOLVE Game on Students Learning*

Both participants agreed that the game impacted positively on the students' learning. The game enhances students' interaction and collaboration in class activities. The correct solution for each question is provided in the game which helps students to critically analyze their approach to solve the given problem statement. Students can collect points and win medals and cups from the game which motivates them to finish exercises.

The game is available online which motivates students to complete exercises at their own convenience and time. Moreover, the game enhances students' problem solving skills, and logic capabilities of the programming domain.

5.2.3. *Strengths of the PROSOLVE Game*

Participants discussed that the game is very simple to use and user-friendly. Students' like to play the game so often even at their home, since it is online and available all the

time, and that it can be accessible even through their mobile devices. The students started competing with each other using the game indirectly helping the instructors in their job. The game based application helps students to do the study by collaborative learning. Students used to get ready with their area of learning by practicing the game before the class. So students get lots of knowledge about the topic of class before its beginning and it help the instructor a lot. The teacher can use the time more effectively and more deep knowledge can be covered within a short period of lecture time. Students started helping each other, in case of doubts regarding any topic of their study and explain each other to clear their doubts by peer study. It helps to improve the interaction among students and they are coming up with better communication skills. This application promotes their logical thinking in solving a problem.

The game is a good supporting teaching tool which promotes practice among students in the IP course. Moreover, it provides question bank to the instructors.

5.2.4. *Weaknesses of the PROSOLVE Game*

There is no background image while playing different levels of the game and also the color patterns and interface remains almost the same throughout the application.

At the beginning of the IP course, students were not familiar with the technical terms presented in the solution of the given problem statement. The game should provide glossary of terms for novice programmers.

5.2.5. *Suggestions for Further Improvement in the PROSOLVE Game*

Database connectivity should be included as an added feature in this game, so that top record holders should be available to all users to induce a spirit of healthy competition among students. Data mining tools should also be included in the application, so that the instructor can make use of common mistakes made by students, error-corrections, future prediction and there by doing better knowledge management.

Participants suggested putting background images in the game. Moreover, they proposed different color combinations and interfaces for different levels of the game.

6. Discussion

The PROSOLVE game was prepared and offered in the IP course to enhance the problem solving skills of novice programmers. The survey was implemented to collect students' feedback and semi-structured interviews were conducted to collect instructors' opinion about the game. Overall, the students and instructors appreciated the game in the teaching and learning process of the IP course.

Students perceived that the game helped them to focus on problem solving skills. Some studies concluded that IP courses pay more attention on syntax and semantics of the programming language compared to the problem solving strategies (Iqbal and Harsh, 2013; De Raadt, 2008). This study helps students to focus on problem solving strategies without much worry about the syntax of the programming language.

The game promotes algorithmic thinking by offering all the questions related to the teaching topics of the course by using problem solving strategy (pseudo code). Students have to devise the solution of the given problem statement by using pseudo code technique which helps them to understand the problem statement in a better way. This finding is consistent with Malik *et al.* (2019) and Sohail *et al.* (2019).

The game covers different teaching topics of the IP course. Students were not just focus on how to design an algorithm but the game also gave them basic understanding of different programming structures. This process promotes deep-learning of the programming domain for students.

The game was offered for the whole semester because it covers most of the teaching topics of the course. This process promotes practice among students. Malik (2016b) mentioned that 'practice is considered an important step in grasping the precise concepts of computer programming for novices' (p. 1).

The game discourages programming shortcut. Webster (1994) discussed that in most cases novice programmers attempt to solve the problem statement directly into the computer program (Problem Statement → Code) which promotes programming shortcut. This study emphasizes the students to follow proper programming process (Problem Statement → Solution plans → Computer program).

Results show that the game helped students in learning all the teaching topics of the course. Each topic is offered as a separate level in the game which helps students to focus on that topic. A number of questions is offered for each topic which also promotes deep-learning of that topic.

Most of the students agreed that the game motivates and engages them in learning computer programming. Prensky (2012) categorized university students who were born since 1980 as 'digital natives'. The technology can play an important role to motivate students in the learning process and engages them as active learners.

The game helps students in developing their interest in the course. This process affects the attrition rates (dropout and failure) in the course which helps computer science and related departments to maintain their students' enrolment. Wiedenbeck *et al.* (2004) concluded that: 'Decisions about majoring in computer science and related fields are often determined by a student's success or failure in the introductory course. If a student drops out, fails, or passes with a struggle, that student is unlikely to enroll for a follow-on course'. (p. 97)

The game is web-based which helps students to finish exercises at their own convenience and time. Students collect points and win medals and cups for their correct solutions against the given problem statements. This process also urges students to practice more questions and acquires deep learning of the programming domain.

Most of the students agreed that the game engages them as active learners in programming activities. This process supports students' cognitive gains and promotes students' affective engagement in the IP course.

Students also appreciated the idea to introduce games in the learning process of advanced courses of programming. Games bring fun and competition elements which attract more attention of students in the learning process. These games should achieve specific goals and cover all the six categories of the Bloom's taxonomy.

IP instructors appreciated the game in the teaching and learning process of the IP course. They considered the game as an additional supporting teaching tool and good alternative of traditional pen and paper learning approach. This mode of learning provides students more access, flexibility and opportunities in learning programming concepts.

IP instructors agreed that the game brings fun element and competition among students in the course. Moreover, the game promotes interaction and collaboration among students in class activities. The game provides correct solution of the problem statement which helps students to get quick feedback of their solutions.

7. Conclusions

The PROSOLVE game was introduced in the introductory programming (IP) course to enhance problem solving skills of novice programmers. Students' feedback was collected by implementing the survey and semi-structured interviews were conducted to collect instructors' opinion about the game.

The result shows that the game helped most of the students in understanding the programming concepts, structures and problem solving strategies. Moreover, the game supports students' cognitive engagement, gains, and affective engagement in the IP course.

Students practice the problem solving strategy for the whole semester because the game covers most of the teaching topics of the IP course. This process helps students to pay equal attention on the programming knowledge (syntax and semantics) and the problem solving strategies.

The game also motivates and engages students in learning the basic concepts of programming domain. Most of the students supported the idea that games should also be introduced in advanced courses of programming.

The game brings fun and competition elements which attract more attention of students in the learning process. Students can win medal, cup and earn points while moving from one level to another level in the game.

IP instructors appreciated the game and considered it as an additional supporting teaching tool in the IP course. Moreover, they considered the game as good alternative of traditional pen and paper learning approach in attracting students' interest in the programming domain.

References

Akcaoglu, M. (2016). Design and Implementation of the Game-Design and Learning Program, *TechTrends*, 60 (2), 114–123, `https://doi.org/10.1007/s11528-016-0022-y`

Battistella, P.E., Wangenheim, C.G.V., Wangenheim, A.V., Martina, J.E. (2017). Design and Large-scale Evaluation of Educational Games for Teaching Sorting Algorithms. *Informatics in Education*, 16 (2), 141–164, DOI: 10.15388/infedu.2017.08

Barnes, T., Richter, H., Chaffin, A., Godwin, A., Powell, E., Ralph, T., Matthews, P., Jordan, H. (2007). The Role of Feedback in Game2Learn, *CHI 2007*, 1–5.

Barnes, T., Chaffin, A., Powell, E., Lipford, H. (2008). Game2Learn: Improving the motivation of CS1 students. *Proceedings of the 3rd international conference on Game development in computer science education*, 1–5, Miami, Florida.

Becker, T. (2010). The Character of Successful Trainings with Serious Games. *International Journal Of Emerging Technologies In Learning (IJET)*. `http://online-journals.org/i-jet/article/view/1498`. Accessed16 October 2018.

Bureau of Labor Statistics, U.S. Department of Labor 2015, *Occupational Outlook Handbook, 2014–15 Edition, Computer Programmers*, Accessed 28 August 2015
`http://www.bls.gov/ooh/computer-and-information technology/computerprogrammers.htm`

Chaffin, A. Doran, K., Hicks, D., Barnes, T. (2009). Experimental evaluation of teaching recursion in a video game. In *Proc. ACM SIGGRAPH Symposium on Video Games*, 79–86.

De Raadt, M. (2008). Teaching programming strategies explicitly to novice programmers, PhD thesis, University of Southern Queensland, Australia, retrieved June 2013, USQ ePrints.

Eagle, M., Barnes, T. (2009). Experimental evaluation of an educational game for improved learning in introductory computing. *SIGCSE,* 41(1), 321–325.

Gunter, G. A., Kenny, R. F., Vick, E. H. (2008). Taking educational games seriously: using the RETAIN model to design endogenous fantasy into standalone educational games. *Educational Technology Research and Development*, 56(5/6), 511–537.

Guzdial, M., Soloway, E. (2002). Log on education: teaching the Nintendo generation to program. *Communications of the ACM*, 45(4), 17–21.

Iqbal, S., Mathew, R., Tawafak, R.M., Khan, I. (2019). Gender Difference in Perceiving Algorithmic Thinking In An Introductory Programming Course, *EDULEARN19 Proceedings*, pp. 8246–8254. doi: 10.21125/edulearn.2019.2042

Iqbal, S., Harsh, O. K. (2013). A self-review and external review model for teaching and assessing novice programmers. *International Journal of Information and Education Technology*, 2(3), 120–123.

Kinnunen, P., Malmi L. (2006). Why students drop out CS1 course?. *Proceedings of the second international workshop on Computing education research*, ACM: ICER '06, 97–108, USA.

Lahtinen, E., Ala-Mutka, K., Järvinen, H.M. (2005). A study of the difficulties of novice programmers. *Proceeding of ITiCSE'05*, ACM:Monte de Caparica.

Lee, M.J., Ko, A.J. (2011). Personifying Programming Tool Feedback Improves Novice Programmers' Learning, *Conference on International Computing Education Research (ICER)*, August 8–9, Providence, Rhode Island, USA, 109–116.

Li, F.W.B., Watson, C. (2011). Game-based concept visualization for learning programming, Proceedings of the third international *ACM workshop on Multimedia technologies for distance learning*, 37–42, December 01-01, 2011, Scottsdale, Arizona, USA.

Malik, S.I., Shakir, M., Elbow, A., Ashfaque, M.W. (2019). Promoting algorithmic thinking in an introductory programming course, *International Journal of Emerging Technologies in Education*, 14 (1).

Malik, S.I. (2018). Improvements in Introductory Programming Course: Action Research Insights and Outcomes, *Systemic Practice and Action Research*, 31(6), 637–656.
`https://doi.org/10.1007/s11213-018-9446-y`

Malik, S.I., Coldwell-Neilson, (2017a). A model for teaching an introductory programming course using ADRI, *Education and Information Technologies*, 22(3), 1089–1120. `https://doi.org/10.1007/s10639-016-9474-0`

Malik, I.S., Coldwell-Neilson, Jo. (2017b). Impact of a new teaching and learning approach in an introductory programming course, *Journal of Educational Computing Research*, SAGE, 55(6), 789–819, DOI: 10.1177/0735633116685852

Malik, S.I., Mathew, R., Hammood, M.M. (2017c). PROBSOL: A web-based application to develop problem solving skills in introductory programming, *In proceedings of 1st International conference AUE*, Dubai,

UAE

Malik, I.S. (2016a). *Role of ADRI model in teaching and assessing novice programmers*, PhD thesis, Deakin University, `http://hdl.handle.net/10536/DRO/DU:30088862` Accessed 15 Oct 2018

Malik, I.S. (2016b). Enhancing practice and achievement in introductory programming using an ADRI editor, *In Proceedings of the IEEE International Conference on Teaching, Assessment and Learning for Engineering*, 32–39, IEEE, Thailand, 7–9 December, DOI:10.1109/TALE.2016.7851766

Malliarakis C., Satratzemi M., Xinogalos S. (2014) Educational Games for Teaching Computer Programming. In: Karagiannidis C., Politis P., Karasavvidis I. (eds) *Research on e-Learning and ICT in Education*. Springer, New York, NY

Maragos, K., Grigoriadou, M. (2011). Exploiting TALENT as a Tool for Teaching and Learning, *The International Journal of Learning*, 18(1), 431–440.

Microsoft Corporation (2012). *A National Talent Strategy: Ideas for Securing U.S. Competitiveness and Economic Growth*, retrieved on 28 August, 2015
`http://www.microsoft.com/en-us/news/download/presskits/citizenship/MSNTS.pdf`

Muratet, M., Torguet, P., Viallet, F., Jessel, J.P. (2010). Experimental feedback on Prog&Play, a serious game for programming practice, *Eurographics*, 1–8.

O'Kelly, J., Gibson, P. (2006). RoboCode & problem-based learning: A non-prescriptive approach to teaching programming. *ACM SIGCSE Bulletin*, 38(3), 217–221.

Paliokas, I., Arapidis, C., Mpimpitsos, M. (2011). PlayLOGO 3D: A 3D interactive video game for early programming education. *Third International Conference on Games and Virtual Worlds for Serious Applications*, 24–31.

Phelps, A., Bierre, K., Parks, D. (2003). MUPPETS: multi-user programming pedagogy for enhancing traditional study, *Proceeding of the 4th conference on Information technology education*, October 2003, Lafayette, Indiana, USA, 100–105.

Piteira, M., Haddad, S. (2011). Innovate in Your Program Computer Class: An approach based on a serious game. *OSDOC – Open Source and Design of Communication Workshop*.

Prensky, M.R. (2012). *From digital natives to digital wisdom: Hopeful essays for 21st century learning*. 1st Edition. SAGE.

Shuhidan, S.M. (2012). Probing the minds of novice programmers through guided learning, PhD thesis, Accessed July 2013, RMIT University: Australia.

Sohail, I.M. (2019). Assessing the teaching and learning process of an introductory programming course with Bloom's Taxonomy and Assurance of Learning (AOL), *International Journal of Information and Communication Technology Education*, 15 (2), DOI: 10.4018/IJICTE.2019040108.

Sohail, I.M., Mathew, R., Al-Nuaimi, R., Al-Sideiri, A., Coldwell-Neilson, J. (2019) Learning problem solving skills: Comparison of E-Learning and M-Learning in an introductory programming course, *Education and Information Technologies*, 24 (5), `https://doi.org/10.1007/s10639-019-09896-1`

Sykes, E.R. (2007). Determining the Effectiveness of the 3D Alice Programming Environment at the Computer Science I Level', *Journal of Educational Computing Research*, 36(2), 223–244.

Theodoraki, A., Xinogalos, S. (2014). Studying Students' Attitudes on Using Examples of Game Source Code for Learning Programming. *Informatics in Education*, 13 (2), 265–277,
DOI: `http://dx.doi.org/10.15388/infedu.2014.07`

Watson, C., Li, F.W.B. (2014). Failure rates in introductory programming revisited, *Proceedings of the ITiCSE*, ACM, Uppsala, Sweden

Webster, M. (1994). *Overview of programming and problem solving*, Merriam-Webster's Collegiate Dictionary, Tenth Edition, Accessed October 20, 2018:
`computerscience.jbpub.com/vbnet/pdfs/mcmillan01.pdf`

Wiedenbeck, S., LaBelle, D., Kain, V.N.R. (2004). Factors Affecting Course Outcomes in Introductory Programming, *Proceedings of 16th Workshop of the Psychology of Programming Interest Group*, Carlow, Ireland, 97–110

Yadin, A. (2011). Reducing the dropout rates in an introductory programming course, *ACM Inroads*, 2(4), 71–76

Yusoff, A., Crowder, R., Gilbert, L., Wills, G. (2009). A Conceptual Framework for Serious Games, *Proceedings of 9th IEEE International Conference on Advanced Learning Technologies (ICALT 2009),* 21–23, 15–17 July 2009 doi: 10.1109/ICALT.2009.19.

Zingaro, D. (2015). Examing interest and grades in computer science 1: A study of pedagogy and achievement goals', *Transactions on Computing Education*, 15(3), ACM

Zualkernan, I.A. (2006) A framework and a methodology for developing authentic constructivist e-Learning environments. *Educational Technology & Society*, 9 (2), 198–212.

R. Mathew is PhD degree holder in Computer Science with a post-graduate degree in Information Technology from India. 25 years of teaching experience [India, UAE, Oman] in the field of I.T. and also with industrial experience as a Computer Programmer. Member of PhD Thesis Evaluation Panel for V.T.U (Visvesvaraya Technological University, Karnataka). Currently working as Software Engineering Coordinator and Faculty Member in Buraimi University College [BUC], Oman. Also worked in BUC as Assistant Head of I.T Department for more than 5 years. Published and presented many papers in various International Journals and Conferences. Research interests in: Data Mining and Knowledge Management, Database Management Systems, Web Site Designing, Cloud Computing, E-Learning, Algorithmic Development for Novice Programmers.

S.I. Malik obtained his PhD in Computer Science from Deakin University, Australia. He has fifteen years of teaching experience, five of which was in Australia. At present, he has been working as an Assistant Professor at Information Technology Department, Buraimi University College, Oman (Academic collaboration with California State University, Northridge, USA) since February 2007. His research interest includes: Algorithmic thinking, Computer Education, Technology in Education, and E-Learning.

R.M. Tawafak got her Bachelor degree from Al-Mousel University with an expert system of bacteria classification project. She got with the fifth honour degree in Computer Science. She worked in Software engineering department in Al-Mustansryia University in Iraq for four years. She worked in Oman College of Management and Technology for two years. Currently, she is a lecturer in information technology department in Buraimi University College, Oman more than twelve years. Most of her publications were indexed under Scopus. Her current research interests include e-learning, Technology Adoption & Acceptance, Academic Performance.