The Use of Games on the Teaching of Programming: A Systematic Review

Thiago Reis da Silva[1], Taina Jesus Medeiros[1], Eduardo Henrique da Silva Aranha[1]

[1]Department of Computer Science and Applied Mathematics – DIMAp/PPgSC
Federal University of Rio Grande do Norte – UFRN/Brazil
thiagoreis@ppgsc.ufrn.br, tainamedeiros@ppgsc.ufrn.br,
eduardoaranha@dimap.ufrn.br

Abstract. Background: Programming teaching is a difficult task due to the course complex nature. Moreover, there are negative stereotypes associated to programming courses, because it often fails to stimulate students to understand. **Aim**: Investigate the efficacy of digital games as a tool to aid the process of teaching and learning programming. **Method**: We conducted a systematic review study to find out how effective is the use of games on programming classes. Three bases of electronic data, thirteen conference and two magazines were researched to find relevant studies. **Results**: After applying the inclusion and exclusion criteria of the 6518 primary studies, 29 were included in this study. **Conclusion**: 97% of the studies report that the use of games is an effective tool for teaching and learning programming.

Keywords: Games, Programming, Systematic Review.

1 Introduction

There are many difficulties faced by novice programmers in computing courses, mainly during the process in learning of programming. There are many reasons for that [1], either by requirement of logical/mathematical thinking, or even by different speeds of students' learning. Learning computer programming has been highlighted as a hard and complex task. This is demonstrated by the high dropout rates school leavers [2] [3] on related courses.

Many efforts have been made by researchers and educators in order to overcome the difficulties found on programming teaching. The use of digital games is a method that is being explored [4] [5], providing interactive and ludic moments to the novice programmers. However, this use is still restricted to isolated initiatives and it is not known how and to what extent the games contribute to the programming teaching process.

Thus, the aim of this work was to perform a Systematic Literature Review (SLR), in order to investigate the use of digital games as a tool for assisting the process of teaching programming. SLR is a research into a phenomenon of interest that produces detailed and specific results by means of a content and quality analysis of the material

researched [6]. SLR was used to remove, catalog, analyze and synthesize data from many papers published in conferences and international journals which address the theme of this review.

This paper is structured as follows: first, we present the research method used (Section 2) and then, we present the general results of this review (Section 3). Finally, we draw some conclusions and present the final considerations (Section 4).

2 Method

The SLR is structured based on the original guidelines proposed by [6]. Regarding to this, the aim was to determine the effectiveness of using digital games as a tool to assist beginning students to learn how to code and analyze research and practical experiences about this subject between 2009 and 2013.

2.1 Research Questions

The SLR performed in this study aims to answer the questions presented in Table 1. The first column shows the research question code, which will be referenced throughout this paper. The second column shows their descriptions.

Table 1. Research Questions

Questions	Description
RQ1	Which programming languages are being taught by studies that make use of games as teaching tools?
RQ2	What studies are being performed by researchers who investigate the use of games on programming teaching?
RQ3	What is the scale (number of attendees) of the studies that are being conducted by researchers?
RQ4	Do the reported studies indicate effectiveness in the use of games for teaching programming?
RQ5	What schooling levels are being contemplated by the studies?
RQ6	The approaches are based on Distance Learning or traditional face-to-face contact?
RQ7	What skills and competencies are being exploited?
RQ8	What benefits and limitations are being reported?

2.2 Searching Process, Inclusion and Exclusion Criteria

The searching process was composed of automatic and manual searches in databases, conferences and journals. This strategy was considered appropriate after pilot studies performed during the drafting of the SLR protocol. Three electronic databases were researched during the SLR, all of them identified as potentially useful in previous studies [7] [8]. The electronic databases were: ACM, CiteSeerX and IEEExplore.

Manual researches in conference proceedings also were made. Thirteen conferences were identified as potentially relevant: *Technical Symposium on Computer Science Education* (SIGCSE), *Conference on International Computing Education Research* (ICER), *Conference on Foundations of Digital Games* (FDG), *Transactions on Computing Education* (TOCE), *Conference on Innovation and Technology in Computer Science Education* (ITiCSE), *International Conference on Digital Game and Intelligent Toy Enhanced Learning* (DIGITEL), *Workshop on Network and System Support for Games* (NetGames), *Symposium on Computacional Intelligence and Games* (CIG), *Conference on Information Technology Education* (SIGITE), *Symposium on Interactive 3D Graphics and Games* (I3D), *International Conference on Software Engineering* (ICSE), *International Symposium on Empirical Software Engineering and Measurement* (ESEM) and *International Conference on Evaluation and Assessment in Software Engineering* (EASE).

Also, the *Journal Computer and Education* (JCE), *Journal of Computing Sciences in Colleges* (JCSC) and *International Journal of Educational Research* (IJER) were examined. The results of this search process are shown in Table 2.

Table 2. Bases included data.

Sources		Amount	1st Selection (Titles and Keyword)	2nd Selection (Abstract and Conclusion)		
				Excluded		Included
				Not Relevant	Duplicate	Primary Studies
String	**IEEE**	147	18	11	0	7
	ACM	212	27	24	1	2
	ScienceDirect	213	2	1	0	1
Conferências	**SIGCSE**	521	11	3	0	8
	ICER	86	8	6	0	2
	FDG	122	3	1	0	2
	TOCE	101	8	7	0	1
	ITiCSE	319	6	3	0	3
	DIGITEL	131	2	1	0	1
	NetGames	76	0	0	0	0
	CIG	286	0	0	0	0
	SIGITE	200	1	1	0	0
	I3D	100	0	0	0	0
	ICSE	967	1	0	0	1
	ESEM	83	0	0	0	0
	EASE	123	0	0	0	0
Journal	**JCE**	1386	3	3	0	0
	JCSC	1195	5	4	0	1
	IJER	250	0	0	0	0
Total		**6518**	**95**	**65**	**1**	**29**

After many executions of different searching strings applied on electronic databases, we found one suitable for the SLR:

teaching AND programming AND games

This searching string was formed after analyzing the keywords and relevant literature titles found during a preliminary search of papers in electronic databases, conferences and journals described earlier. To build the searching string for this review, keywords that define this thematic were selected: (1) *Game*, (2) *Teaching* and (3) *Programming*. We were careful to include in the searching string some variants terms (*education*, *code*, *program*) and other grammatical inflections.

A three-steps approach was used in the searching process. In the first step, all retrieved studies were evaluated in order to identify those relevant to answer the research questions. A total of 6.518 studies were analyzed. After reading the titles, abstracts and keywords, this initial set was reduced to 95. During this selection, Inclusion Criteria (IC) and Exclusion Criteria (EC) were applied (see Table 3).

Table 3. Inclusion and Exclusion Criteria.

Inclusion Criteria	Exclusion Criteria
IC1: Papers reporting the use of games on the teaching of programming for students of basic, technical and higher education; **IC2**: In the case of journals or conferences reporting the same study, only the more recent paper was included; and **IC3**: Publications between 2009 and 2013.	**EC1**: Publications were excluded if their main focus was not on the use of games for programming teaching; **EC2**: Papers proposing one approach or describing use of games on programming teaching, but without any evaluation; **EC3**: Documents which are available in the form of abstracts or presentations and secondary studies (systematic mappings and literature reviews); **EC4**: Publications not written in English; and **EC5**: Duplicated papers.

In the second step, papers abstracts and conclusions were read, and again, IC and EC were applied to the 95 candidate papers. The result was a subset of 29 primary studies (63 excluded based on exclusion criterion C2, 1 based on exclusion criterion C3, 1 based on criterion C5 and 1 duplicate paper.). It is important to emphasize that, in some cases, the reading of other parts of papers was done in order to proceed to the selection. In the third step, the 29 studies were read in full. The references of these papers are available on <http://goo.gl/jLJh3n>.

2.3 Quality Evaluation

Each relevant paper was evaluated by its quality. This quality evaluation process was performed during the data extraction phase and it ensured that the included stud-

ies had a valuable contribution to SLR. It was considered the 11 criteria for quality evaluation discussed by [9]:

- **QA1**: Is the study related to empirical research or experience reports based on reports or in the experts opinion?
- **QA2** There is a clear definition about the research goals?
- **QA3**: There is a suitable description of the context in which the research was conducted?
- **QA4**: The research planning was suitable to address the research goals?
- **QA5**: Data extraction strategy was suitable to the research goals?
- **QA6**: There was a control group in which we could compare treatments?
- **QA7**: Data were collected addressing the research questions?
- **QA8**: Data analysis was sufficiently accurate?
- **QA9**: The relation between researcher and attendees was considered in a suitable degree?
- **QA10**: There is a clear indication of the results?
- **QA11**: Is the value of study for research or practice?

The first two of these criteria were used to exclude papers that do not clearly indicate the research goals. This represents the minimum quality threshold that has been observed during SLR. The remaining criteria are intended to determine the accuracy, analysis and credibility of the research methods employed, as well as the relevance and quality of each study for this review. The answers to each question were tabulated and assigned a value of 1 ("Yes") or 0 ("No"). In order to test the procedure validity for quality evaluation, two additional reviewers received the same samples found in the papers and were invited to assess the quality based on the same criteria defined for quality evaluation. There was no disagreement about quality evaluation in these papers. See results in Table 4.

2.4 Data extraction

In this phase, the following data were collected on the selected studies: study goals; study method (i.e., controlled experiment, experience report, case study, among others); number of attendees involved in studies; how data were collected and analyzed during the study; schooling which the survey was applied (i.e., high school); type of programming language that is being taught; the results, conclusions and problems found; cognitive abilities worked in studies; effectiveness use of games as a method for teaching of programming; and the study quality evaluation. All data were extracted by a reviewer, while other two examined the overall outcome of the studies. These results were then compared and no significant issues were found. All the extracted data were stored in a spreadsheet.

3 Results

This section summarizes the results of the study. Thus, in Subsection 3.1 presents the evaluation of the quality of primary studies, the next subsection the answers to research questions, and finally, in Subsection 3.3 is described limitations and threats of the study.

3.1 Quality evaluation of the primary studies

The results of this quality evaluation are presented in Table 4. A quality score was attributed for each paper. All papers included in the review were based on empirical research or presenting experience reports [QE1], and the goals were clearly defined [QE2]. From 29 studies, 18 offered some context description in which the research was conducted [QE3], while 26 had an appropriate research planning [QE4].

Table 4. Quality Evaluation Results.

Studies	QA1	QA2	QA3	QA4	QA5	QA6	QA7	QA8	QA9	QA10	QA11	Total
S1	1	1	0	1	1	0	0	0	0	1	1	**6**
S2	1	1	1	1	1	1	1	0	0	0	1	**8**
S3	1	1	1	1	1	0	0	1	0	1	1	**8**
S4	1	1	1	1	1	0	0	1	0	1	1	**8**
S5	1	1	0	1	0	0	0	0	1	1	1	**6**
S6	1	1	1	1	1	0	0	1	0	1	1	**8**
S7	1	1	1	1	1	1	1	1	1	1	1	**11**
S8	1	1	1	1	1	1	1	1	1	1	1	**11**
S9	1	1	1	1	1	1	1	1	1	1	1	**11**
S10	1	1	0	1	0	0	0	1	0	1	1	**6**
S11	1	1	0	1	1	0	1	0	0	1	1	**7**
S12	1	1	0	1	1	0	0	1	0	1	1	**7**
S13	1	1	0	1	1	0	0	0	1	1	1	**7**
S14	1	1	1	1	1	0	1	0	1	1	1	**9**
S15	1	1	0	1	1	0	0	1	1	1	1	**8**
S16	1	1	0	1	1	0	0	1	0	1	1	**7**
S17	1	1	0	1	1	0	0	0	0	1	1	**6**
S18	1	1	0	0	1	0	0	0	1	1	1	**6**
S19	1	1	1	1	1	0	1	0	1	1	1	**9**
S20	1	1	1	1	0	0	1	1	0	0	1	**7**
S21	1	1	1	1	1	1	1	1	1	1	1	**11**
S22	1	1	1	1	1	1	1	0	0	1	1	**9**
S23	1	1	1	1	1	0	1	0	0	1	1	**8**
S24	1	1	1	0	1	1	1	0	0	1	1	**8**
S25	1	1	1	1	1	0	1	0	0	1	1	**8**
S26	1	1	1	1	1	0	1	0	0	0	1	**7**
S27	1	1	1	1	1	0	1	0	0	1	1	**8**
S28	1	1	1	1	1	0	1	0	0	1	1	**8**

S29	1	1	0	0	0	1	1	1	1	0	1	**7**
TOTAL	**29**	**29**	**18**	**26**	**25**	**8**	**17**	**13**	**11**	**25**	**29**	**/**

The analysis of Table 4 shows that the included studies had an appropriate extracting strategy to the research context [QE5], but the most part did not use a control group that could compare the results [QE6]. In 17 studies, the data were collected in a way that addressed the research questions [QE7]. However, the data analysis was considered sufficiently accurate only in 13 studies [QE8]. Relation between researcher and attendees was considered suitable in 12 studies [QE9]. On the other hand, only 4 studies did not provide a clear indication of results [QE10]. Finally, all studies present a value for the research or practice [QE11]. 4 studies included in the review obtained maximum score of 11 points [S7, S8, S9 and S21]. The lower score reached by the papers was 6 points [S1, S10, S17 and S18]. The average score in quality of the studies included in the review is 7,9 points. Finally, it was considered that all papers had enough quality to support the analysis of our research questions.

3.2 Research Questions

The answers to the research questions described in Subsection 2.1 are discussed below. In Table 5, we have the publications by country statistics, according to the country of affiliation of authors. Here, we see that most research activity in this area is in the United States, accounting for 59% of research, followed by England (7%), Malaysia (7%) and China (7%). The other countries in Table 5 corresponding to 4% each. A summary of the data extracted from each study are presented in Table 6.

Table 5: Papers by country.

Country	Papers	Study
United States	18	E1, E3, E4, E8, E9, E10, E13, E14, E15, E16, E17, E21, E22, E24, E26, E27, E28, E29
England	2	E11, E19
Malaysia	2	E12, E20
China	2	E23, E25
South Africa	1	E2
Greece	1	E6
Brazil	1	E7
Finland	1	E18
Czech Republic	1	E5

Table 6. Summary of included studies.

Studies	Programming Language	Type of study	Scale participants	Games are effective?	Level	Emphasis
S1	Java	Interviews	17	Yes	University	Classroom
S2	-	Experimental Evaluation	-	Yes	University	Classroom
S3	-	Controlled	30	Yes	Various	Classroom

		Experiment				
S4	-	Quasi-experiment	14	Yes	University	Classroom
S5	-	Experience Report	100	Yes	University	Classroom
S6	Java	Controlled Experiment	-	Yes	High School	Classroom
S7	C++	Case Study	30	Yes	High School	Classroom
S8	C#, Visual Basic e F#	Case Study	50	Yes	University	Distance
S9	Java	Case Study	18	Yes	High School	Classroom
S10	Python	Controlled Experiment	230	Yes	University	Distance
S11	-	Questionnaire	25	Yes	University	Classroom
S12	-	Controlled Experiment	30	Yes	University	Classroom
S13	Java	Case Study	18	Yes	High School	Classroom
S14	Java	Experience Report	325	Mixed		Classroom
S15	Java	Workshop	12	Yes	High School	Classroom
S16	-	Case Study	322	Yes	High School	Classroom
S17	Python	Case Study	88	Yes	High School	Classroom
S18	C #	Case Study	233	Yes	High School	Classroom
S19	-	Case Study	25	Yes	University	Classroom
S20	-	Controlled Experiment	10	Yes	University	Classroom
S21	-	Case Study	13	Yes	University	Classroom
S22	Java	Experience Report	-	Yes	University	Classroom
S23	Visual Basic	Quasi-experiment	146	Yes	University	Classroom
S24	C#	Questionnaire	14	Yes	University	Classroom
S25	-	Controlled Experiment	133	Yes	High School	Classroom
S26	C++	Controlled Experiment	92	Yes	University	Classroom
S27	C	Workshop	8	Yes	High School	Classroom
S28	-	Workshop	26	Yes	University	Classroom
S29	Java	Experience Report	9	Yes	University	Classroom

[RQ1] Which programming languages are being taught by studies that make use of games as teaching tools?

Analyzing the studies included in SLR, we found 7 different types of programming languages that are been used. Java was the main language used, found in 8 studies [S1, S6, S9, S13, S14, S15, S22 and S29]. Three papers discuss the use of C# language [S8, S18 and S24]. Visual Basic [S8 and S23], Python [S10 and S17] and C++ [S7 and S26] were reported in 2 papers each. C [S27] and F# [S8] were both described by a single paper. The other studies did not show any programming languages used. The studies emphasized the use of software designed to teach programming principles. This includes the use of Greenfoot [S9, S13, S15, S17 and S28], Alice [S1, S13, S14 and S16], Scratch [S16 and S25], Blocky [S3] and GameMaker [S29].

The use of robotics for programming teaching principles was also reported. The use of Lego Mindstorms addressing theory and practice of programming introduction using robotics was employed in two studies [S2 and S3]. The LighBot is an environment of a simple educational programming that allows novice programmers to learn the basics of programming in a simulated robot, and its use was highlighted in [S19]. In [S10], it is presented a game called Gidget, in which students play through a series of levels, finding and fixing robot defects.

[RQ2] What studies are being performed by researchers who investigate the use of games on programming teaching?

The use of the case study was the most employed method, being reported in 10 studies, which corresponding to 35% of the total [S6, S7, S8, S9, S13, S16, S17, S18, S19 and S21]. The use of controlled experiments was described in 6 studies [S3, S10, S12, S20, S25, S26], corresponding to 21%. Report experience was employed in 4 papers [S5, S14, S22 and S29)], representing 14%. The use of workshops was reported in 3 studies [S15, S27 and S28], corresponding to 10%. Quasi-experiment [S4 and S23] and Questionnaires [S11 and S24] were used in 2 studies, completing a total of 14%. In addition, the Experimental evaluation [S2] and Interviews [S1] also occurred corresponding to a total of 6%. These data reveal that in 18 studies were performed using a empirical research method (case study, controlled experiment and quasi-experiment) and in 11 studies, no empirical research method was used.

[RQ3] What is the scale (number of attendees) of the studies that are being conducted by researchers?

The studies vary from studies of small scale that contained 8, 9, 10 and 13 attendees, to studies of larger scales with 100, 146, 233, 322 people. The largest sample was 325. In 27 studies it was reported the exact number of students who answered the survey. On the other hand, 2 studies [S6 and S2] did not report the number of the attendees involved.

[RQ4] Do the reported studies indicate effectiveness in the use of games for teaching programming?

Analyzing all the studies included in SRL, it is possible to produce a classification about reporting the use of digital games as an effective method of teaching and learning of programming. From the 29 papers, 28 (97%) reported that the use of games is effective for teaching of programming concepts, and a single study offers a mixed result [S14], i.e., it can show positive and negative results, but this study did not re-

ported negative points. No study reported that use of games is considered inefficient for teaching of programming.

[RQ5] What schooling levels are being contemplated by the studies?

The papers were classified according to the educational level in which the research was made. It is possible to notice that the most of the studies are focused on the higher education (55%) [S1, S2, S4, S5, S8, S10, S11, S12, S19, S20, S21, S22, S23, S24, S26 and S29]. In contrast, 39% of the analyzed studies have focused on the high school [S6, S7, S8, S9, S13, S14, S15, S16, S17, S18, S25 and S28]. Only 3% are intended to elementary education [S27]. This review also shows that 3% of the papers are intended to teaching in all levels (elementary, high school and higher education) [S3]. These data show that 43% of the studies are being designed to basic education (elementary and high school) and 57% to higher education, showing that the teaching of programming at the basic level is a research trend.

[RQ6] The approaches are based on Distance Learning or traditional face-to-face contact?

With respect to this aspect, 98% of the studies are intended for teaching on-campus (face-to-face) and only 2% for distance learning. With the growth of distance learning, it suggests a research opportunity for this educational field.

[RQ7] What skills and competencies are being exploited?

In the studies reported, just a few of them clearly have skills and competencies being exploited. However, two skills were found: cognitive and social. Cognitive skills reported in the studies were: players building algorithms during the game-play, designing their own solutions (collaboration) [S11]; players usually use logic condition in order to achieve a higher score in the game and also when they want to create reusable solutions (establish logical conclusions) [S7] [S16]; players can debug their solution to detect errors in their logic (solving problems) [S10]. In social skills, the characteristics presented were: dealing with rules; cooperation and development [S11] [S18] [S20] [S35] and [S47]. Regarding to competencies, all studies have mentioned, such as: creativity, structuring of thought, responsibility, curiosity, confidence and team work.

[RQ8] What benefits and limitations are being reported?

Using the results obtained by SLR, it was observed that some benefits and limitations were reported. The benefits were: the use of games motivates and enhances the teaching of programming, increasing power of concentration in students and the school pass rate, as well as a decrease in the number of leavers in algorithm subjects; use of workshops as a ludic and attractive ways of teaching-learning process for algorithms; use of ludic tools for teaching of programming – such as Scratch, GameMaker, Blocky, Alice and Greenfoot – as a way to make more dynamic the teaching-learning process. [E19] pointed out that high school students had a significant increase in computing knowledge, also a growth interest in this field and the prospect of doing computing courses in the future. [E11] emphasized that higher education students reported that the use of games for teaching of programming in introductory subjects is efficient to assist new novice programmers.

In general, the difficulties faced by students do not reflect something unusual, because all computing students have the same problem, especially at the beginning of a course: difficulties to abstract and understand problems, or to specify and encode a solution. The difficulties identified by the papers were: to understanding the problems and the ability to think logically. However, in [E30], the authors draw attention to some challenges, such as: technology used for developing games is not suitable; not everyone likes games; and the large time for developing attractive games to students.

We can also identify some weaknesses regarding to programming teaching-learning in the general context, such as:

- The students do not know how to solve problems through programming: we believe that the most important cause for these difficulties that many novice programmers feel in learning of programming is the lack of skills to solve common problems; this involves problem understanding and logical reasoning;
- Many students do not have enough logical and mathematical knowledge: we believe that mathematical knowledge is very important for programming and it was possible to find studies, such as [12], that evidences the relation between programming skills and mathematics;
- Programming requires a high abstraction level: programming learning requires abstraction skills, generalization and critical thinking, among others;
- The lack of motivation in learning of programming: many students do not have enough motivation to study, because there is a connotation that is extremely negative associated with programming that goes from student to student.

These problems are being addressed and resolved by the researchers by the use of digital games as an attractive and motivating factor for programming teaching-learning.

3.3 Limitations of this study

The procedures used in this study have deviated in aspects presented in the guidelines [6] for some specific points:

- The research was organized in a manual and automatic searching process in a specific sets of journals and conferences;
- A single researcher selected the studies, based on their titles and abstracts;
- A single researcher arranged the data, while the others have verified data extraction as suggested by [8].

The first topic implies the possible absence of relevant studies. The automatic searching only happened in 3 databases, deemed acceptable by [10] and a manual searching happened in 13 conferences presented as relevant in [7]. In particular, it may be noticed the missing of relevant papers published in other journals and national conferences. This threat was controlled by choosing the main databases, events and journals in this field.

Regarding to second topic, as the number of papers analyzed during the preselection is high. There is a chance that relevant papers have passed unnoticed in

analysis of the titles, abstracts and keywords and also in the searching string. This threat was controlled by making use of experienced researchers in the field with available time to make a careful selection.

The third and last point means that some data collected may be incorrect. Control of this threat was carried out by a careful analysis of each study, since in this third point, there were relatively few primary studies. Control of this threat was carried out carefully by an experienced researcher in SRL and analysis of these data types, since this third point there were few primary studies.

3.4 Lessons learned from the SLR

Many lessons can be reported after the completion of this review. [10] discussed the experiences of a PhD student in conducting a SLR. [7] added other discussions and lessons learned. Including observations that: conduct a SLR is a time consuming and laborious process; and suggests that the development of a protocol helps to perform the review.

From the lessons learned in conducting the presented study, it is possible to agree with the authors that the SLR conducting a large-scale, independently, as part of a doctoral research project, is a time-intensive process. In fact, the study presented here took about ten months to get done. This time period was reported as an average for PhD students [11].

The lessons learned from the experience of conducting the study independently also allow several observations to be made that may help other students, even on PhD, mastering and undergraduate who decide to perform an SLR. Then, these lessons are:

- Define clearly the research questions and the search procedure;
- Perform various tests of string to adapt to each search engine's;
- A single researcher can develop a protocol for SLR with the accompaniment of an experienced researcher; and
- The use of appropriate inclusion criteria and exclusion can reduce the number of papers accepted for the study and thus has a direct correlation with the effort exerted.

4 Final Considerations

Many analyzes can be performed based on the results on this review. Regarding to quality index, we believe that this number is good (average of 7.9/11), mainly because 18 of 29 papers included in the review implement some empirical study method. These studies provide good scores concerning the quality evaluation criteria used for this review.

Eight research questions were created in order to determine the efficiency of using games in learning of programming and providing a broad view of the research theme. Many results and trends concerning the teaching of programming using games may be indicated with results including observations that:

- The Java programming language has been the most frequently adopted in researches;
- Case studies and controlled experiments are the most commonly methods reported and used to evaluate the effectiveness of the games intervention;
- The attendees number who took part in the research to evaluate the effectiveness of using games in teaching of programming varies from small to large samples;
- Half of the work is being applied to higher education and a considerable amount of these studies focuses on-campus teaching (face-to-face).

The data also show that the research in the games field for programming teaching has been increasing in the last two years. In fact, the number of papers published has increased significantly in 2012 and 2013, having approximately twice the number of publications on previous years. Another relevant point is the discrepancy between the number of articles published in the contexts of classroom education and distance. These data clearly showed the need for research that addresses the teaching of programming in distance education.

Through the research conducted it was possible to verify the application of the use of games in teaching programming. The results in this application indicate that the use of digital game is an effective tool for teaching program, i.e., 97% of the reported studies. In addition to the results of SLR, there was a discussion that describes the lessons learned in conducting a SLR from the perspective of a PhD student. Some contributions were made in order to help other students who decide to develop an SLR. Another contribution of this work is the SLR itself, which will serve as the starting point for several other researches through primary studies and reported results.

In all the analyzed studies, the term "motivation" is often used to justify the developed researches. The motivational factor is the reason why the potential of digital games is being investigated in order to increase the interest of students and reduce the number of leavers in subjects related to programming. We can deduce that the search for innovations to promote motivation and interest of the students is a constant in the analyzed studies. However, after a more detailed analysis on the justifications of these studies, it was found that under the term "motivation", different goals are included. The goals identified were: improve certain skills, i.e., logical reasoning; students' engagement in interactive and ludic activities; maximize the learning of a particular content, i.e., conditional structures and repetition; and group learning. In general, the SLR results indicate that the use of games is an effective tool for programming teaching. This is observed in 97% of the selected studies.

Acknowledgments

The authors thank the CAPES for partial support to this research, the National Institute of Science and Technology for Software Engineering (INES), the CNPq, and in particular to the Postgraduate Program in Computer Systems (PPgSC/UFRN) for all offered infrastructure.

References

1. Steven, S., Betsy, D., Mark, G. Using game development to reveal programming competency. Proceedings of the Foundations of Digital Games – FDG, 2012.
2. Bayliss, J. D. Using games in introductory courses: tips from the trenches. In: Proceedings of the ACM Technical Symposium on Computer Science Education - SIGSE, 2009.
3. Michael E., Tiffany B. Experimental Evaluation of an Educational Game for Improved Learning in Introductory Computing. Proceedings of the ACM Technical Symposium on Computer Science Education - SIGSE, 2009.
4. Sá, E. J. V, Teixeira, J. S. F, Fernandes, C. T. Design learning activities using games as principle for Cooperation. Proceedings of the Brazilian Symposium on Informatics in Education - SBIE. 2007.
5. Cinnamon H., Robin A., Michael P., Kelvin S., John N., David G. Game-Themed Programming Assignments For Faculty: A Case Study. Proceedings of the ACM Technical Symposium on Computer Science Education - SIGSE, 2010.
6. Kitchenham, B. A. Guidelines for performing Systematic Literature Reviews in Software Engineering Version 2.3, Keele University and University of Durham, EBSE Technical Report, 2007.
7. Major, L., Kyriacou, T., Brereton, O. P. Systematic Literature Review: Teaching Novices Programming Using Robots. Proceedings of the International Conference on Evaluation and Assessment in Software Engineering – EASE, 2011.
8. Brereton, P., Kitchenhama, B. A., Budgenb, D., Turnera, M., Khalilc, M.. Lessons from applying the systematic literature review process within the software engineering domain, Journal of Systems and Software 80 (4) (2007) 571–583.
9. Dybå, T., Dingsøyr, T. Empirical Studies of Agile Software Development: A Systematic Review. Information and Software Technology, 50 (9-10), pp. 833–859. 2008.
10. Woodall, P., Brereton, P. Conducting a Systematic Literature Review from the Perspective of a PhD Student (Extended Abstract). *International Conference on Evaluation and Assessment in Software Engineering* – EASE, 2006.
11. Riaz, M., Sulayman, M., Salleh, N., Mendes, E. Experiences Conducting Systematic Reviews from Novices Perspective. Proceedings of the Evaluation and Assessment in Software Engineering – EASE, 2010.
12. Sheard, J.; Simon, S.; Hamilton, M.; Lönnberg, J. Analysis of research into the teaching and learning of programming. *ACM International Computing Education Research* – ICER, Berkeley, 93-104, 2009.