Systematic mapping on the use of games for software engineering education

Diego Castro[1], Diogo Matheus Costa[1], and Cláudia Maria Lima Werner[1]
{diegocbcastro, dmc, werner}@cos.ufrj.br

COPPE/Computer Systems Engineering Program, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

Abstract. Games appeal to all kinds of people and are used for a wide range of purposes, from pure entertainment to educational research. Software Engineering is an area of computing that has much content to explore, distributed in different knowledge areas. Thus, there are many theoretical classes that make learning tiresome. In this scenario, educators and researchers are looking for new methods to engage students in order to facilitate the learning process. Many studies have already been produced using games as a teaching method. However, a recurring mistake when using games as a teaching method is the application of the wrong approach to a context that often hinders student learning. This paper presents a systematic mapping that aims to identify the main ways that games are being used as a teaching method focused on Software Engineering. The results highlight five ways of using games as a teaching method, where Game-based Learning, Digital Game-based Learning and Gamification are the most used ones.

Keywords: Serious Games, Software Engineering, Systematic Mapping.

1 Introduction

Going to college and entering a classroom to study a particular subject becomes a routine in a student's daily life. Over time, this can become a tedious situation where students end up losing interest in the matter. In this scenario, educators are increasingly exploring innovative learning strategies that combine pleasure with education [1]. Trying to make learning something enjoyable can be a great help to students who need to study a subject. It is worth remembering that according to Wolfman [2], people remember 15% of what they hear, 25% of what they see and 60% with which they interact. Since games are mostly interactive, this becomes an excellent opportunity to make students remember what has been taught. Based on this, the use of games in education seeks to make the teaching of a subject something more interactive and more pleasurable [3].

The idea of games was introduced into the academic world, creating a new vocabulary: "serious games", where this instrument provides a framework of entertainment, in which content is incorporated [1]. Squire [4] states that recent studies show that when playing a particular game, people are involved in more complex and challenging learning activities than most college tasks. Keeping in mind that games can be used to develop new skills and engage in more complex operations than those presented in the classroom, it is intended to use the full potential of games to facilitate the learning of Software Engineering (SE).

This paper presents a systematic mapping focused on the use of games as a teaching method in SE, identifying the main ways that games are being used for this purpose. The remainder of this paper is presented as follows: Section 2 presents a brief background on the use of games as a teaching method; Section 3 describes the research method used in the systematic mapping; Section 4 shows the results that were found; and Section 5 concludes with the final remarks.

2 Background

SE is an area of computing that seeks to create the specifications of a project and assist in the development and maintenance of systems [5]. SE is related to all stages of a software product and, therefore, it is necessary to have a good knowledge of the process that it addresses. However, it has many concepts and subdivisions related to it (i.e., knowledge areas), such as requirements, software quality, project management, among others [6].

To be able to have competent professionals in the future, they must have a good understanding of the concepts that are taught at the university. Connolly, Stansfield, and Hainey [7] claim that SE has been described as a "perverse problem", where different points make it difficult to be taught, such as few questions with clear answers, among others. In addition, Navarro and Van Der Hoek [8] affirm that there is a large gap in traditional SE education techniques, where "students are exposed to various concepts and theories of SE in lectures but have few opportunities to put these ideas into practice". Claypool and Claypool [9] argue that, currently, many SE projects lack the fun factor to engage students in learning. To reduce this teaching difficulty, different concepts have been employed to try to teach it otherwise. One is the use of games.

Caillois [10] define a game as an activity that is fun for the user and has an unpredictable result that is inside something fictitious and does not generate anything productive. According to Salen, Tekinbaş, and Zimmerman [11], a game is a system in which players participate in an artificial, rule-defined conflict that concludes a quantifiable result. Therefore, in general, games can be defined as activities that use an abstract world through decisions, actions, and rules, which aim at a recreational activity, in the form of distraction or amusement [12].

Games that have functions beyond entertainment, such as teaching, training, informing, or assisting in something, are called serious games. According to Adams and Dormans [13], the term serious game was designed in recognition that games can be used for purposes other than entertainment. Although there is no standard definition for serious games, this work adopts the following description: serious game is a game developed with a purpose beyond entertainment, offering opportunities for the user to try new approaches with safety, low cost, and without consequences [13].

3 Research Method

Systematic mapping is a method that makes aggregation of information based on studies related to a specific research topic [14], based on a structured and repeatable protocol that explores and categorizes studies and provides an overview [15]. The mapping to be presented was carried out in the period from 2018-12-15 to 2019-03-02 and followed the protocol proposed by Kitchenham [16], covering the articles published until the end of 2018.

The **protocol implementation** procedure followed three well-defined phases: Execute the search string in Scopus as recommended by other studies [17, 18]; apply the inclusion/exclusion criteria based on the title, abstract and full text; and perform the tasks from the previous phase for the articles found in the snowball procedure that will be described in detail in the next section. The **inclusion criteria** selected for this study were: the article must be in the context of using games for ES teaching; the article must provide data to answer at least one of the research questions; and the article should be written in English. The **exclusion criteria** were: Book chapters, conference call; and studies that can not be fully accessed.

3.1 Research Questions

- **Q1:** What game definition was used?
- **Q2:** What is the main advantage / motivation of the use of games to teaching software engineering?
- **Q3:** What is the disadvantage of the use of games to teaching software engineering?
- **Q4:** What are the research areas that the game intends to teach?
- **Q5:** What is the main characteristic of the game used?

3.2 Search String

The definition of the search keywords was made based on the PICO strategy [19], using three of the four levels.

The search string was defined by grouping keywords of the same domain with the logical operator “OR” and grouping the three domains with the logical operator “AND”. In the first instance, the search string formed by the three PICO levels returned a result of 4,428 articles. Since the number of papers returned was relatively large, it was decided to use a date filter, searching only for documents that were published within the last five years, as performed by Jiang, Zhang, Gao, Shao, and Rong [20]. The keywords, PICO structure and the search string can be seen in Table 1.

To validated the search string, an ad-hoc literature review was performed by selecting two knowledge control articles that can be seen in Table 1. Using this kind of control, the validation is performed by executing the search string and it should return the selected control articles.

Table 1. Descriptive table with search string and keywords.

Pico	**Synonyms**
Population	Software engineering
Intervention	Tutoring, teach*, instruction, discipline, schooling, education*, mentoring, course, learn*, train*, syllabus
Comparison	Not applicable
Outcome	Game*, gami*, play*, “serious games”, edutainment, “game based learning”, simulation
Search String	
Control Article	(1) Software engineering education and games: A systematic literature review; (2) SimulES-W: A collaborative game to improve software engineering teaching
Filter	(LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014))
TITLE-ABS-KEY ((“Software engineering”) AND (tutoring OR teach* OR instruction OR discipline OR schooling OR educat* OR mentoring OR course OR learn* OR train* OR syllabus) AND (game* OR gami* OR play* OR “serious games” OR edutainment OR “game based learning” OR simulation) AND (LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014)))	

4 Results

The search string returned a total of 1,229 papers. Applying the inclusion and exclusion filters, this number dropped to 37 papers. As there is no perfect search string, and the work was only performed at Scopus, the snowball procedure was used to minimize the lack of other search bases [17]. This procedure aims to find new references based on the articles that were cited in the papers that were read (backward snowballing) and in relation to the papers that referred to the article read (forward snowballing) [17]. In the backward snowballing process, 382 articles were identified; applying the filters, this number dropped to only 2. In forward snowballing, 311 articles were identified, and 5 more articles were included. After all the search process, 44 papers were included. Table 2 shows the detailed analysis of the articles and Table 3 lists all articles that were included in the review, demonstrating which questions each article was able to answer.

Table 2. Analysis of the articles

	Main study		**Snowballing backward**		**Snowballing forward**	
Activity	**Result**	**Number of papers**	**Result**	**Number of papers**	**Result**	**Number of papers**
First Execution	1229 added	1229	382 added	382	311 added	311
Repeated Papers	32 withdraw	1197	36 withdraw	346	78 withdraw	233
Papers in another language	12 withdraw	1185	17 withdraw	329	51 withdraw	182
Remove conference / workshops	57 withdraw	1128	0 withdraw	329	0 withdraw	182
Remove books	43 withdraw	1085	21 withdraw	308	4 withdraw	178
Remove by title	638 withdraw	447	236 withdraw	72	112 withdraw	66
Remove by abstract	379 withdraw	68	64 withdraw	8	51 withdraw	15
Papers not found	0 withdraw	68	1 withdraw	7	2 withdraw	13
Remove by full paper	31 withdraw	37	5 withdraw	2	8 withdraw	5
Total papers included	**37 documents**		**2 documents**		**5 documents**	
Extracted Papers	**44 documents**					

4.1 Findings’ Summary

Q1: What game definition was used? Video games have been in our lives for over 50 years, quickly becoming one of the most important, lucrative, and

Table 3. Traceability matrix.

Title	Year	Q1	Q2	Q3	Q4	Q5
A serious game to support the ISO 21500 standard education in the context of software project management	2018	X	X		X	X
What can go wrong in a software project? Have fun solving it	2018	X	X	X	X	X
Games for Requirements Engineers: Analysis and Directions	2018	X	X		X	X
How to design gamification? A method for engineering gamified software	2018	X	X	X		X
Classutopia: A serious game for conceptual modeling design	2018	X	X		X	X
SimulES-W: A collaborative game to improve software engineering teaching	2018	X	X		X	X
Experiences of using a game for improving learning in software requirements elicitation	2018	X	X	X	X	X
Gamification Applied in the Teaching of Agile Scrum Methodology	2018	X	X		X	X
Gamification in Software Testing: A Characterization Study	2018	X	X		X	X
Gamification in Software Engineering Education: An Empirical Study	2017	X	X			X
Teaching ISO/IEC 12207 software lifecycle processes: A serious game approach	2017	X	X	X	X	X
Integrating serious games as learning resources in a software project management course: The case of ProDec	2017	X	X		X	X
Game elements in a software engineering study group: A case study	2017	X	X		X	X
Teaching software testing concepts using a mutation testing game	2017	X	X	X	X	X
A case study of software engineering methods education supported by digital game-based learning: Applying the SEMAT Essence kernel in games and course projects	2017	X	X		X	X
Developing and implementation of decision-making games for business education of engineering students	2017	X	X	X	X	X
Simkan: Training kanban practices through stochastic simulation	2017	X	X		X	X
Deploying a gamification framework for software process improvement: Preliminary results	2017	X	X	X		X
Coverage of ISO/IEC 29110 project management process of basic profile by a serious game	2017	X	X		X	X
Coverage of the ISO 21500 Standard in the Context of Software Project Management by a Simulation-Based Serious Game	2017	X	X	X	X	X
Gamification of software testing	2017	X	X		X	X
An agile software engineering process improvement game	2016	X	X		X	X
GSDgame: A serious game for the acquisition of the competencies needed in GSD	2016	X	X	X		X
The effects of students' motivation, cognitive load and learning anxiety in gamification software engineering education: a structural equation modeling study	2016	X	X	X		X
A serious game to promote object oriented programming and software engineering basic concepts learning	2016	X	X		X	X
Methodology to construct educational video games in software engineering	2016	X	X			X
Improving programming education through gameful, formative feedback	2016	X	X	X		X
Coverage of ISO/IEC 12207 software lifecycle process by a simulation-based serious game	2016	X	X		X	X
Ace that game: Educating students to gamified design thinking	2016	X	X	X	X	X
SCRUM-X: An interactive and experiential learning platform for teaching scrum	2016	X	X		X	X
Proposal to Teach Software Development Using Gaming Technique	2016	X	X		X	X
Project management game 2D (PMG-2D): A serious game to assist software project managers training	2015	X	X	X	X	X
The issues of adopting simulation games in software engineering classes	2015	X	X			X
Experimental evaluation of a serious game for teaching software process modeling	2015	X	X		X	X
Gamification of software engineering curriculum	2015	X	X	X		X
Gamification proposal for a software engineering risk management course	2015	X	X		X	X
Examining the effectiveness of 2D and 3D online environment in enhancing students' learning of software engineering	2015	X	X			X
Anukarna: A software engineering simulation game for teaching practical decision making in peer code review	2015		X		X	X
Implementing gamification techniques into university study path - A case study	2015	X	X			X
InspectorX: A game for software inspection training and learning	2014	X	X		X	X
Research on teaching gamification of software engineering	2014	X	X	X		X
Evaluating GameDevTycoon for teaching Software Engineering	2014	X	X		X	X
A general framework for software project management simulation games	2014	X	X		X	X
StrateJect: An Interactive Game for Project Management Experiential Learning	2014		X		X	X

influential ways to entertain [21]. A game is an activity between two or more independent decision-makers trying to achieve their goals in some limited context. A more conventional definition would say that a game is a context with rules between opponents trying to achieve goals [22]. According to Laskowski [23], a game is a non-serious and voluntary activity, intentionally separated from the real world. An activity that absorbs the player fully and intensely and is not related to any benefit. A game requires the player to play at a specific time and place and to do so under the established order and rules [23].

One of the first serious games to be developed was Army Battlezone, a project developed by Atari in the early 1980s that was created to train soldiers on a battlefield [24]. However, for Gannod, Burge, and Helmick [25], the term serious game was first used only in 2002 in a release of a military training simulator called “America’s Army”, created by the US Army and distributed for free over the Internet.

Serious games are designed in such a way that the primary purpose is not to entertain the user but to train him/her in a particular area. This does not mean that the game cannot be fun, but the entertainment derived from the game is designed to educate so that the player’s learning experience becomes fun [26]. Serious games are designed for a different purpose than just entertaining the user [22, 27]. The adjective “serious” indicates that their goal is more than just fun and that they are designed to educate, train, or inform users [28]. For Maxim, Kaur, Apzynski, Edwards, and Evans [29], serious games make use of the artistic means of games to convey a message, teach a lesson, or provide an experience.

According to Ramírez-Rosales, Vázquez-Reyes, Villa-Cisneros, and De León-Sigg [21], a serious game with a didactic purpose is one in which some kind of knowledge is obtained through playing, facilitating the teaching and learning process.

In this scenario, this question was added to find out the game definitions that were used, aiming to identify the main ways (i.e., approaches) that games are being used as a teaching method focused on Software Engineering. In general, five ways of using games as a teaching method were found: Game-based Learning; Digital Game-based Learning; Game Development Based Learning; Simulation; and Gamification. Based on the collected data, Game-based Learning, Digital Game-based Learning and Gamification are the most used ones.

Garcia, Pacheco, León, and Calvo-Manzano [30] use the term Game-based Learning (GBL) to represent the application of games for educational purposes. For them, GBL can be defined as applying the principles of traditional games in real-life situations, to get students’ attention and encourage knowledge creation. Thus, the concept of GBL mostly refers to the use of games to support teaching and learning [30, 31]. GBL can be understood as any use of a game for educational purposes. However, it has a branch called Digital Game-based Learning (DGBL) that comprises games in their digital versions [32].

One more approach is Game Development Based Learning, that represents the activity of developing games as a practical learning experience. Games represent an eminent domain of students, thus facilitating the understanding of requirements. Games are attractive, motivating, and involve students in their creative process. Finally, the varied complexity of games can provide educators with a wide range of options for classroom tasks and projects [33].

Another approach that has been found as a method that can improve the current way of teaching is the use of Simulation, that can be understood as attractive approaches within a risk-free environment [34]. Simulation games mimic a given situation and bring a more realistic experience [35]. Nassal [36] believes that simulation games are an alternative tool to gain experience in a controlled and risk-free manner. Simulation games also let users try out various strategies at once.

Finally, another approach related to enriching training and teaching through games is the phenomenon of Gamification, which, according to Calderón, Ruiz, and O'Connor [22] can be understood as the use of game elements and game design techniques in non-playful contexts. Unkelos-Shpigel [37] adds by saying that this technique is defined as integrating game mechanics into non-playful environments to increase engagement, loyalty, fun, and can even stimulate user participation and contribution. For Herranz, Colomo-Palacios, and Al-Barakati [38] gamification is the use of game elements in other contexts to alter and inspire people's behavior. The intention is to inject fun, play, and passion into tasks and processes [38, 39]. Gamification has attracted the attention of practitioners and researchers as a way to use game-based mechanics, aesthetics, and thinking to motivate people, promoting learning to increase end-user engagement and change behaviors in many contexts, such as teaching [22].

Q2: What is the main advantage / motivation of the use of games to teaching software engineering? Using games as a reinforcement tool to teach skills can be a very beneficial strategy for students. They have proven to be a useful tool to complement conventional learning methods. Games allow visualizing concepts that could be perceived as too abstract. They also help to familiarize the students with the knowledge and techniques that may be tedious to study, offering a cycle of challenges and rewards that drives the learning experience [26, 40, 41].

Many benefits are linked to the use of games in education, such as increased collaboration and competition, immediate feedback, the possibility of reflecting on the results achieved, and the transfer of content so that learning is an integral part of education [42]. Another advantage attributed to serious games is the ability of the user to assume real roles [43]. As training tools, serious games increase students' conceptual knowledge, increase task completion, student's confidence, and improve knowledge retention [44].

SE education lacks practical experience, which is essential for understanding problems and especially their solutions [36]. It is crucial that these experiences are based on real scenarios so that the student can make a good use of the content. Many areas of higher education are difficult to practice due to their long duration and the potential for severe damage. However, games represent an alternative tool for gaining experience in a controlled and risk-free manner [36]. Playing does not take as much time and care as in a real project, and in case of failure, the game can be restarted. Games have a significant advantage for teaching; that is their repeatability. They can be reset at any time, allowing students to learn through their failures [45].

A game is a visible and physical representation of a problematic space, a captured mental model that can be repeated. As such, they are places to test new ideas and experiment with established theories; repeat the activities as many times as necessary; areas where time and space can be contracted or expanded; places where it is only acceptable to try different things and where it is possible to learn more by failure than by success [36].

Simulation increases the opportunity to learn from failure without any loss in reality, but in a very timely manner, and can bring engagement and motivation on the part of students. One of the main advantages that simulation can bring is feedback, in which, the student knows if he/she is doing a particular task right or wrong, seriously influencing the learning effects. Also, simulation games promote numerous cognitive benefits for students, such as motivation, visual effects, experimentation, self-efficacy, self-monitoring, problem-solving, and critical thinking skills [46].

Finally, the gamification approach can have a positive impact on increasing user motivation and engagement concerning a particular behavior [32].

Q3: What is the disadvantage of the use of games to teaching software engineering? Few papers have reported problems. In this scenario, only 7 of 44 studies mentioned disadvantages.

The limited use of serious games in formal education may be related to the issues surrounding the use of leisure games, such as a view where games can be addictive, unproductive and may teach misconceptions [26]. Another very relevant point about not using games as a teaching method is that players usually learn through repetition, patterns, and exploration, which contrasts with learning discrete amounts of information, as in college.

To create an educational game is not enough to use a game that performs an activity to teach something. To effectively teach and retain practice, students need to be more active and being provided with better instructional methods in which they participate in higher-level cognitive tasks [24]. Another problem with using games is that they are just a representation of how the real world works. Therefore, it is potentially dangerous to the players leave the game environment with the belief that the effective strategies employed in the game are directly transferable to the real world [47].

Gamification explores different game elements. However, some of these elements may cause problems if not used correctly. Herranz, Colomo-Palacios, and Al-Barakati [38] stated that it is not easy to manage and that there is a danger of misinterpreting the conceptualization and improper implementation of game elements, and, finally, they stated that gamification is not easy to manage because there is a risk of misinterpretation through users, which occurs in unexpected results, such as the introduction of unfair competitiveness, demotivation generated by scoreboards, among other problems. Morschheuser, Hassan, Werder, and Hamari [48] complement by saying that games in education are challenging to design different reasons, such as games are complex, multifaceted and therefore difficult to create; they aim to affect behavior and not just entertain, which implies understanding a range of (motivational) psychology and requires appropriate skills in the development team.

Q4: What are the research areas that the game intends to teach? As already mentioned in Section 1, SE has many subdivisions (i.e., knowledge areas) to be taught, such as requirements, software quality, project management and more. In this scenario, this question was added to find out which SE area are most explored through games.

To answer this question, the identified studies were classified. However, some studies did not have a well-defined SE area. Therefore, this kind of study was classified as others. Although the work references only 44 papers (i.e., selected documents), different studies addressed more than one SE area. The frequency of each area is presented as follows: Project Management (13 out of 44); Software Engineering (11 out of 44); Software Testing and Software Quality (5 out of 44); Software Development, Software Process and Agile Methodology (4 out of 44); Software Requirement (3 out of 44); Software Modeling (2 out of 44); Risk Management and Software Inspection (1 out of 44). Based on the collected data, the mostly addressed SE area was Project Management (13 studies).

Q5: What is the main characteristic of the game used? Different games and ways to use them as a teaching method have been analyzed, each one has its characteristics, which give "life" to the game and make it productive or not. Among the main features observed, the ones that stand out most are a real-world representation, levels or phases, constant feedback, and the use of graphs that catch the user's eye.

When building an education game, some contextual characteristics should be taken into account. Games must represent the real world complexity, emphasizing authentic tasks and encouraging experiences and reflections through collaboration, competition, social negotiation, simulations and feedback [42]. Another essential information that must be taken into account is the creation of dependent phases, where the player can gradually learn and build up his/her knowledge [49].

There are also features that reference the game's interface and gameplay. Among these characteristics stand out mainly the narratives and graphics that can bring the right motivation. An attractive game must have a minimalistic graphic, displaying easy-to-interpret buttons, lists, and menus. Similarly, game elements should be natural to interpret using some familiar game components such as life bars, dialog windows, score, avatars, scoreboard, leaderboard, checkpoints, sprites, boss fights, missions, badges, and rewards in general [27].

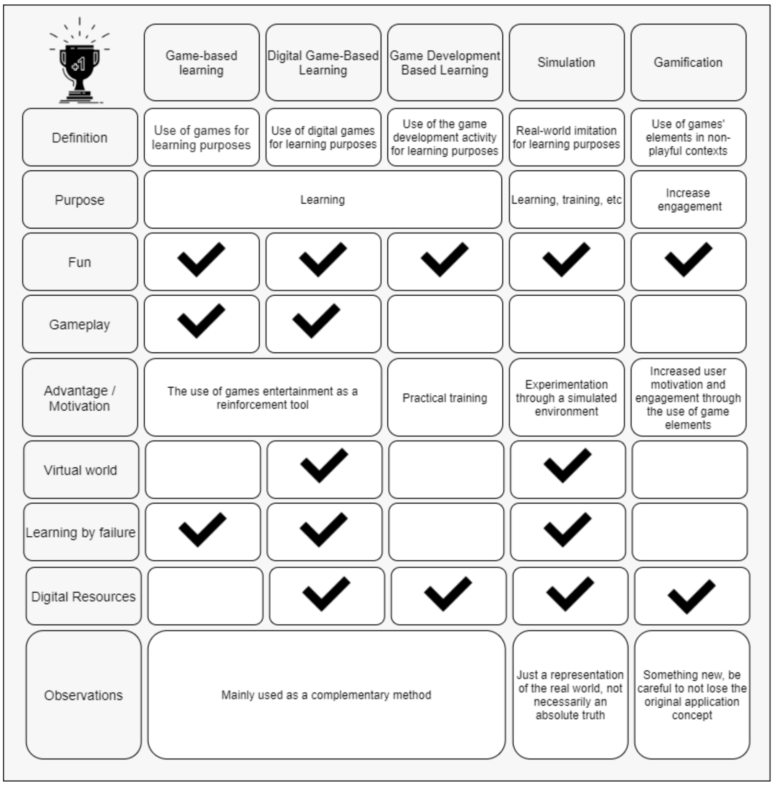
One of the most common concepts in games is the idea of levels. In educational games, they are used to divide content that should be presented in a sequential manner, which can be aggregated with the concept of time [21, 22]. Another important feature that was applied in games was the use of the level map and a help session, where the player can go back to the part that he/she had the most difficulty in solving, but now having extra help [21].

In this scenario, the Gamification approach can provide several features to non-playful context, such as levels, points, badges, avatars, leaderboard, progress bar, title changes, and status changes [29, 39]. In this scenario, the use of online plataforms was observed, such as Moodle to manage content [37].

Finally, it was also identified the main languages used to develop educational games, which were Java, JavaScript, C++ and C Sharp. In addition, the use of the Unity 3D game engine was explored when related to 3D programming because of the ease of exporting to multiple platforms (e.g., PC, Mac, web and mobile) [8, 21, 24, 28, 41].

Summary Result. This study identified different games and characteristics that sought to increase motivation and engagement through fun. In most studies, games are used by participants with minimal or no knowledge in the subject to be taught. In this case, it is expected that these games have resources to introduce certain concepts in a way that the player feels more engaged to continue learning the subject, as well as to help in the practical training.

In general, five ways of using games as a teaching method were found. In this scenario, Figure 1 shows a table summarizing the collected data to facilitate the reader to be able to choose the method that best suits his/her teaching purpose. Based on it, the reader can see the purpose of each approach and its main characteristics, and from that, make his/her interpretations and find out which one to use.



	Game-based learning	Digital Game-Based Learning	Game Development Based Learning	Simulation	Gamification
Definition	Use of games for learning purposes	Use of digital games for learning purposes	Use of the game development activity for learning purposes	Real-world imitation for learning purposes	Use of games' elements in non-playful contexts
Purpose	Learning			Learning, training, etc	Increase engagement
Fun	✓	✓	✓	✓	✓
Gameplay	✓	✓			
Advantage / Motivation	The use of games entertainment as a reinforcement tool		Practical training	Experimentation through a simulated environment	Increased user motivation and engagement through the use of game elements
Virtual world		✓		✓	
Learning by failure	✓	✓		✓	
Digital Resources		✓	✓	✓	✓
Observations	Mainly used as a complementary method			Just a representation of the real world, not necessarily an absolute truth	Something new, be careful to not lose the original application concept

Fig. 1. Findings' summary

5 Final Remarks

The presented work aimed to identify the main ways that games are being used as a teaching method in SE. To achieve this goal, a systematic mapping was performed based on five questions. These questions sought to find out which methods were most used based on their advantages, disadvantages, elements and characteristics.

Based on the collected data, it was possible to observe that the most used methods are Game-based Learning, Digital Game-based Learning and Gamification, but there are other ways to use games in education, such as Simulation and Game Development Based Learning.

Some limitations can be identified by performing a critical analysis of the work. The used search filters represent a scope limitation. In addition, the ad-hoc categorization of the SE knowledge areas also limit the work. In this scenario, the SWEBOK knowledge areas could be used.

The present work is part of a more extensive research focused on the use of games as a teaching method for Software Reuse (SR). Initially, a research focused on the use of games for teaching SR was conducted. However, no results were found. Since SR is one of the SE knowledge areas, it was decided to generalize the research by searching the use of games for teaching SE. Thus, it is expected that it will be possible to use the collected data to support the creation of a game to teach SR.

Acknowledgment. The authors would like to thank CAPES and CNPq for their financial support.

References

1. Ritterfeld U, Cody M, and Vorderer P. Serious games: Mechanisms and effects. Routledge, 2009.
2. Wolfman DE. Criando em multimídia. Rio de Janeiro: Campus 1994 (in portuguese).
3. Ludens H. A Study of the Play Element in Culture. Trans. by RFC Hull.(London, 1949) 1955:168.
4. Squire K. Changing the game: What happens when video games enter the classroom? Innovate: Journal of online education 2005;1.
5. Pressman RS. Software engineering: a practitioner's approach. Palgrave Macmillan, 2005.
6. Sommerville I, Arakaki R, and Melnikoff SSS. Engenharia de software. Pearson Prentice Hall, 2008.
7. Connolly TM, Stansfield M, and Hainey T. An application of games-based learning within software engineering. British Journal of Educational Technology 2007;38:416–28.
8. Navarro EO and Van Der Hoek A. Software process modeling for an educational software engineering simulation game. Software Process: Improvement and Practice 2005;10:311–25.

9. Claypool K and Claypool M. Teaching software engineering through game design. In: *ACM SIGCSE Bulletin*. Vol. 37. 3. ACM. 2005:123–7.
10. Caillois R. Les jeux et les hommes. Le masque et le vertige. Editions Gallimard, 2015 (in french).
11. Salen K, Tekinbaş KS, and Zimmerman E. Rules of play: Game design fundamentals. MIT press, 2004.
12. Xexéo G, Carmo A, Acioli A, et al. O Que São Jogos? 2017 (in portuguese).
13. Adams E and Dormans J. Game mechanics: advanced game design. New Riders, 2012.
14. Keele S et al. Guidelines for performing systematic literature reviews in software engineering. Tech. rep. Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.
15. Steinmacher I, Chaves AP, and Gerosa MA. Awareness support in distributed software development: A systematic review and mapping of the literature. Computer Supported Cooperative Work (CSCW) 2013;22:113–58.
16. Kitchenham B. Procedures for performing systematic reviews. Keele, UK, Keele University 2004;33:1–26.
17. Matalonga S, Rodrigues F, and Travassos GH. Characterizing testing methods for context-aware software systems: Results from a quasi-systematic literature review. Journal of Systems and Software 2017;131:1–21.
18. Motta RC, Oliveira KM de, and Travassos GH. Characterizing Interoperability in Context-aware Software Systems. In: *2016 VI Brazilian Symposium on Computing Systems Engineering (SBESC)*. IEEE. 2016:203–8.
19. Petticrew M and Roberts H. Systematic reviews in the social sciences: A practical guide. John Wiley & Sons, 2008.
20. Jiang S, Zhang H, Gao C, Shao D, and Rong G. Process simulation for software engineering education. In: *Proceedings of the 2015 International Conference on Software and System Process*. ACM. 2015:147–56.
21. Ramírez-Rosales S, Vázquez-Reyes S, Villa-Cisneros JL, and De León-Sigg M. A serious game to promote object oriented programming and software engineering basic concepts learning. In: *2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT)*. IEEE. 2016:97–103.
22. Calderón A, Ruiz M, and O'Connor RV. A serious game to support the ISO 21500 standard education in the context of software project management. Computer Standards & Interfaces 2018;60:80–92.
23. Laskowski M. Implementing gamification techniques into university study path-A case study. In: *2015 IEEE Global Engineering Education Conference (EDUCON)*. IEEE. 2015:582–6.
24. Lino JEN, Paludo MA, Binder FV, Reinehr S, and Malucelli A. Project management game 2D (PMG-2D): A serious game to assist software project managers training. In: *2015 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2015:1–8.

25. Gannod G, Burge J, and Helmick M. Using the inverted classroom to teach software engineering. In: *2008 ACM/IEEE 30th International Conference on Software Engineering*. IEEE. 2008:777–86.
26. Valencia D, Vizcaino A, Garcia-Mundo L, Piattini M, and Soto JP. GSDgame: A serious game for the acquisition of the competencies needed in GSD. In: *2016 IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW)*. IEEE. 2016:19–24.
27. Larenas F, Marín B, and Giachetti G. Classutopia: A Serious Game for Conceptual Modeling Design. In: *SEKE*. 2018:116–5.
28. Calderón A and Ruiz M. Coverage of ISO/IEC 12207 software lifecycle process by a simulation-based serious game. In: *International Conference on Software Process Improvement and Capability Determination*. Springer. 2016:59–70.
29. Maxim BR, Kaur R, Apzynski C, Edwards D, and Evans E. An agile software engineering process improvement game. In: *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2016:1–4.
30. Garcia I, Pacheco C, León A, and Calvo-Manzano JA. Experiences of using a game for improving learning in software requirements elicitation. Computer Applications in Engineering Education 2019;27:249–65.
31. Su CH. The effects of students' motivation, cognitive load and learning anxiety in gamification software engineering education: a structural equation modeling study. Multimedia Tools and Applications 2016;75:10013–36.
32. Uskov V and Sekar B. Gamification of software engineering curriculum. In: *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. IEEE. 2014:1–8.
33. Souza MR, Veado L, Moreira RT, Figueiredo E, and Costa H. Games for learning: bridging game-related education methods to software enginering knowledge areas. In: *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*. IEEE Press. 2017:170–9.
34. Morales-Trujillo ME, García-Mireles GA, and Maslova P. What can go wrong in a software project? Have fun solving it. In: *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE. 2018:995–1004.
35. Peixoto DC, Resende RF, and Pádua CIP. The issues of adopting simulation games in software engineering classes. In: *2015 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2015:1–8.
36. Nassal A. A general framework for software project management simulation games. In: *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE. 2014:1–5.
37. Unkelos-Shpigel N. Ace That Game: Educating Students to Gamified Design Thinking. In: *STPIS@ CAiSE*. 2016:75–82.
38. Herranz E, Colomo-Palacios R, and Al-Barakati A. Deploying a Gamification Framework for Software Process Improvement: Preliminary Re-

sults. In: *European Conference on Software Process Improvement.* Springer. 2017:231–40.

39. Jesus GM de, Ferrari FC, Paula Porto D de, and Fabbri SCPF. Gamification in Software Testing: A Characterization Study. In: *Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing.* ACM. 2018:39–48.
40. Qu WQ, Zhao YF, Wang M, and Liu BQ. Research on teaching gamification of software engineering. In: *2014 9th International Conference on Computer Science & Education.* IEEE. 2014:855–60.
41. Jiménez-Hernández EM, Oktaba H, Piattini M, Arceo FDB, Revillagigedo-Tulais AM, and Flores-Zarco SV. Methodology to construct educational video games in software engineering. In: *2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT).* IEEE. 2016:110–4.
42. Pieper J, Lueth O, Goedicke M, and Forbrig P. A case study of software engineering methods education supported by digital game-based learning: Applying the SEMAT Essence kernel in games and course projects. In: *2017 IEEE Global Engineering Education Conference (EDUCON).* IEEE. 2017:1689–99.
43. Lopes FJR and Petrillo F. SimKan: Training Kanban Practices Through Stochastic Simulation. In: *Brazilian Workshop on Agile Methods.* Springer. 2016:110–21.
44. Pötter H, Schots M, Duboc L, and Werneck V. InspectorX: A game for software inspection training and learning. In: *2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T).* IEEE. 2014:55–64.
45. Szabo C. Evaluating GameDevTycoon for teaching software engineering. In: *Proceedings of the 45th ACM technical symposium on Computer science education.* ACM. 2014:403–8.
46. Calderón A, Ruiz M, and Orta E. Integrating serious games as learning resources in a software project management course: the case of ProDec. In: *Proceedings of the 1st International Workshop on Software Engineering Curricula for Millennials.* IEEE Press. 2017:21–7.
47. Caulfield C, Xia JC, Veal D, and Maj S. A systematic survey of games used for software engineering education. Modern Applied Science 2011;5:28–43.
48. Morschheuser B, Hassan L, Werder K, and Hamari J. How to design gamification? A method for engineering gamified software. Information and Software Technology 2018;95:219–37.
49. Dlamini RS. Examining the effectiveness of 2D and 3D online environment in enhancing students' learning of software engineering. International Journal of Innovation and Learning 2015;17:425–52.