

Educational Games and Learning Effectiveness

Daniela C. C. Peixoto ¹, Rodolfo F. Resende ², Clarindo Isaías P. S. Pádua ²

¹Centro Federal de Educação Tecnológica (CEFET-MG)
Department of Computer - Belo Horizonte, MG - Brazil

²Universidade Federal de Minas Gerais (UFMG)
Department of Computer Science - Belo Horizonte, MG - Brazil

cascini@decom.cefetmg.br, {rodolfo, clarindo}@dcc.ufmg.br

Abstract. *Assessments of educational games have to take into account many aspects, including internal and external game mechanisms. This paper analyses the assessment methods incorporated in a specific Software Engineering educational game, named SPIAL. We used four assessment categories that are commonly found in the literature: aim, implementation, integration and primary type. We observed that the main assessment mechanisms incorporated in SPIAL is the score. Other mechanisms were applied mostly for guiding students during the learning process. In addition, the external evaluations did not consider internal actions of the game and were used mostly to certify the acquired skill. Such findings are in line with the mechanisms used in other Software Engineering educational games. This paper also provides insights for researchers in issues related to the definition of educational game assessment mechanisms and implementation in general, an example is the evaluation of the effective learning compared to the evaluation of the player's perception of learning.*

1. Introduction

Educational games¹ are gaining increasing attention in the literature as an effective tool to enhance the learning and understanding of complex subjects [Vos 2015]. The use of educational games shifts from “learning by listening” to “learning by doing” model of teaching. Many claims have been made about the benefits of using games in the academic context. For instance, educational games can develop students' skills, provide effective learning, enhance students' motivation or change behavior and attitudes [Connolly et al. 2012, Boyle et al. 2016, Calderón and Ruiz 2015, Caballero-Hernández et al. 2017].

Educational games in a Software Engineering curriculum suitably fit as complementary components to lectures, projects, and readings [Carrington et al. 2005, Mandl-Striegnitz 2001]. Our previous experience has demonstrated that it is important to use simulation games complementary to other educational techniques, which should be accompanied by an adequate amount of direction and guidance given to the students [Peixoto et al. 2012, Peixoto et al. 2014]. In the recent years, an increasing number of Software Engineering educational games can be found in the literature, e.g. SimSE [Navarro 2006], The Incredible Manager [Barros et al. 2006] and

¹Although the terminology may vary in the literature, in the context of this study, we refer to educational games as a synonym of serious games.

SimVBSE [Jain and Boehm 2006]. However, despite the potential benefits of using educational games in the Software Engineering courses, there is a lack of substantial evidence on their learning effectiveness [Qian and Clark 2016]. In this context, most of the game evaluations are, thus far, preliminary and informal in nature [Connolly et al. 2012, Calderón and Ruiz 2015, Petri and von Wangenheim 2017]. The researches employ a mixed-method approach, including pre and post domain specific tests, questionnaires, interviews, and video analysis. Usually these methods cover few characteristics of the game environment, varying according to the needs and experiences of its designers [Peixoto et al. 2014]. While there is an excitement about the development and usage of educational games, it is important to conduct more rigorous evaluation in order to justify the investments and understand their real contributions.

Reflecting the continuous interests in educational games, the aim of this paper is to extend our previous research about games evaluation [Peixoto et al. 2014], analyzing the mechanisms used to assess the players' skills. This analysis includes not only the aspects embedded on the game but also external evaluations carried out after playing the game. The assessments mechanisms of SPIAL reflect common elements adopted by other Software Engineering educational games. In general, Software Engineering educational games do not embed learning effects verification on it. This is an important field of research topic, since until now most of Software Engineering educational games only had informally evaluated, using questionnaires based on the students' perception of learning.

The rest of this work is structured in the following way. In Section 2 we define relevant terms and concepts that appear afterwards in the paper. Section 3 describes the methods used to conduct this study. Then, we present the case study analysis in Section 4 and 5. Section 6 discusses the main findings. Finally, we list the conclusions of this study and identify future work in Section 7.

2. Background

In this section, we present the educational game used as a case study. Then, we introduce important concepts related to the assessment methods.

2.1. Software Engineering Educational Game

SPIAL [Kupsch 2012] is a graphical, interactive, and adaptable simulation game (Figure 1). The game goal is to improve the Software Engineering learning, using simulation. Several Software Engineering topics are explored within the context of a Software Process Improvement project. These aspects were based on CMMI-DEV version 1.3 [CMMI 2010]. CMMI was chosen because it is the most widely known SPI (Software Process Improvement) reference model.

SPIAL was developed with the goal to provide students with a more realistic experience in software development processes within the academic environment. SPIAL allows students to practice SPI techniques and the best practices of Software Engineering. SPIAL is a single-player game in which the player takes on the role of a manager of an SPI group in a software development organization. SPIAL's scope covers both a development project and an improvement project. The player is given a process improvement task and he or she can interact with other stakeholders (high level management, project manager, team member, consultant, or customer) represented as non-player characters,

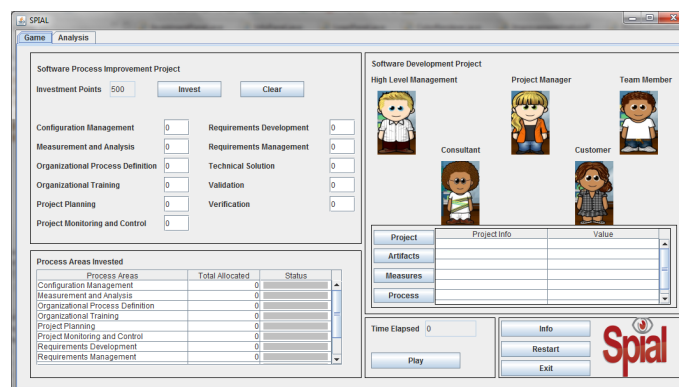


Figure 1. SPIAL Graphical User Interface.

i.e. a character controlled by the computer (see Figure 1). The type of improvement that is required is stated at the beginning of the game and it can include, for example, cost and defect reduction and productivity improvement. In order to complete the task, the player can make investments for improving specific process areas of a software development project. A good investment strategy will result in improvement of process areas and a bigger budget for further investments. The player can visualize project estimations, indications of process areas capability level, and decide in which process areas to invest. During the running development project, the player can visualize the effects of his/her selections on the outcomes (productivity, defect, cost, and time-to-market measures) and, if needed, change his/her investments. The final outcome is a score that represents how close the results are to the initial proposed target. During the game, the non-player characters communicate the effects of the player's actions through bubbles over their heads represented on the interface.

2.2. Learning Assessment

According to the European Commission [Commission 2012], assessment is defined as the “inference about an individuals’ knowledge, skills, attitudes or other capabilities with reference to pre-defined criteria and using one or more assessment methods such as tests, observations, interviews, projects or portfolios”. In educational games, the purpose of including assessment mechanisms is to evaluate the students’ learning process.

Skill assessments can be conducted by different methods [Crisp 2012, Chaudy et al. 2013, Caballero-Hernández et al. 2017]. Usually, these methods are classified into four categories: *aim*, *implementation*, *integration* and *primary type* (see Table 1). The *aim* category is defined as the purpose of including the assessment in the learning environment. It includes diagnostic, formative, summative, and integrative methods. The *implementation* category consists of adopting assessment elements in the game environment. It includes the game scoring, internal and external methods. The assessment *integration* is the approach for conducting the assessment and includes the monitoring of states, quests, use of an assessment model or profile, non-invasive assessment, quizzes and peer assessments methods. Finally, the *primary types* indicates where the assessment is carry out and includes in-process, completion and teacher assessment.

In this work we focused on the elements used to implement the learning assessment during and after game play. We also analyzed the external assessments mechanisms

used by SPIAL.

Table 1. Assessment methods (Extracted from Caballero-Hernández and colleagues [Caballero-Hernández et al. 2017])

<i>Classes</i>	<i>Methods</i>	<i>Description</i>
Aim	Diagnostic	Determine the skills that students possess.
	Formative	Propose improvements to the learn process.
	Summative	Validate and grade the skills
	Integrative	Provide feedbacks in order to support the students' autonomy.
Implementation	Game Scoring	Assess according to the targets achieved or tasks completed.
	Embedded	Assessment is part of the game environment.
	External	Assessment is not part of the game environment.
Integration	Monitoring of states	Each game state is associated with a probability of achieving a learning target.
	Quests	Quests are used to assess the student.
	Use of an assessment model or profile	Student's performance is compared with an existing model.
	Non-invasive assessment	Reduce the stress of students when they are evaluated during the game play.
	Quizzes	Quizzes integrated during the game play.
	Peer assessments	Informal peer assessment.
Primary Types	In-process	Provides an analysis of the player's choices during the game.
	Completion	Indicates the success of the player in completing the game.
	Based on teacher's observation	Teacher observations during and after the game play.

2.3. Educational Game Evaluations

Connolly and colleagues [Connolly et al. 2012] conducted an extensive literature review on computer games and serious games. They identified 129 primary studies reporting empirical evidence about the impacts and outcomes of games with respect to a variety of learning goals. Their findings revealed that playing computer games is linked to a range of perceptual, cognitive, behavioral, affective, motivational impacts, and outcomes. The most frequent outcomes in serious games are knowledge acquisition and content understanding. They also have pointed out the necessity of more rigorous evidence of games' effectiveness and an adequate measurement tool for soft and social skills.

Calderón and Ruiz [Calderón and Ruiz 2015] identified, through a systematic literature review, 102 primary studies describing the procedures, techniques and methods used to assess serious games. They have noticed that a larger number of the educational games were used for training in higher education and questionnaires were the main method used for the evaluation. A large number of studies used both quantitative and qualitative questions, and the most assessed attribute is the learning outcome. This indicates that the main interests, in these primary studies, are to determine effectiveness in terms of knowledge acquisition.

The work of Petri and von Wangenheim [Petri and von Wangenheim 2017] identified, through a systematic literature review, seven approaches to evaluate educational games. They observed that some of the approaches were developed and conducted in an *ad-hoc* way. They collected 52 different quality/sub-quality factors in these researches that have been used to evaluate educational games. In most of the primary studies, learning was the main quality factor used in the evaluation, followed by other 51 factors, including social interaction, challenge, competence and immersion, and fun. They also identified that the majority of these approaches have collected data via questionnaires. The authors concluded that it was needed more investments on research in order to improve the approaches for evaluating educational games.

3. SPIAL Assessments

For our work, we analyzed the assessment elements used in SPIAL and the possible learning effects that they produced on students. We also discuss the application of an external evaluation framework and their coverage considering the learning process. Our discussion is based on the skill categories described on Section 2.2 and on the UGALCO evaluation framework [Peixoto et al. 2014].

We conducted the following activities, described in the next sections: (1) we analyze how SPIAL incorporated the skill assessment elements; (2) we present the external assessments carried out by SPIAL; and (3) we contrast and compare the students' observations with the assessment elements.

4. Embedded Skill Assessment Mechanisms

In order to ease the identification of the assessments elements that were used in the educational game, we evaluated its signs. According to Peirce, signs are anything that stand for something (else) in someone's perspective [Peirce 1992]. The Semiotic Engineering theory classifies the signs in an interactive system into three classes of signs: *metalinguistic*, *static* and *dynamic* [de Souza 2005]. *Metalinguistic signs* are signs that refer to other interface signs. They are instructions, tips, online help, error and informative messages, warnings and system documentation. They are signs that the designer uses to explicitly communicate to users the meanings encoded in the systems and how they can be used. *Static signs* express and mean the system's state, they are, generally, motionless and persistent when no interaction is taking place. They can be perceived (and interpreted) in snapshots of the system's interface before or after interaction occurs. For instance, buttons, text areas and check boxes at a given moment. *Dynamic signs* express and mean the system behavior. Their representations unfold and transform themselves in response to an interactive turn. For example, if we click on the search button the behavior will present the results of a search. This behavior is a *dynamic sign*.

The analysis of the *metalinguistic signs* allows us to identify important elements used in the designers' discourse. The *metalinguistic signs* are used to provide a better understanding of the *static* and *dynamic signs*, and the intended relation among them. To analyze the *metalinguistic signs* the downloading site, dialogs, and messages were inspected in regard to what they conveyed about the system's assessment mechanisms. During the analysis of the *static signs*, important signs were identified at the main interface of the game. The analysis of *dynamic signs* yields more interesting and informative results for our evaluation. With this analysis, it is possible to evaluate the dynamic aspects of the game and it is also possible to see the effects of some of the simulation parameters on the attributes. Table 2 lists the main signs identified in this educational game.

These signs convey important assessment mechanisms of the game environment. Once we analyzed the signs, we observed that the most frequent assessment type found on SPIAL has the formative aim. The formative aim is represented by different types of feedback, such as bubbles speech, messages, and graphs. These feedback mechanisms are useful resources for helping players to understand concepts related to software process improvement. It also supports players' decisions about the game goals, contributing to improve the learning process. In the implementation class we have the score. The score is presented to players at the end of the game and it is calculated according to a

Table 2. SPIAL Signs.

<i>Class</i>	<i>Signs</i>
Metalinguistics	<i>Feedback through speech bubbles</i> : These bubbles convey important information about the effects of the actions and also provide some guidance. <i>Message showing when an organization is immature</i> : Mature organization has data to generate graphs; probably it implements a measurement program. <i>Textual description of the game rules</i> : Students can learn how to improve the score reading the rules used in the game.
Static	<i>Software development organization environment</i> : The game simulates a software organization. <i>Current project status</i> : As in a development company, it is possible to see some project metrics. It is possible to generate graphs and follow some targets proposed by the game.
Dynamic	<i>Feedback mechanisms</i> : the process areas status (whether the capability level is increasing), the project metrics, the process improvement analysis (e.g. “defect was not reduced in 10.0% as expected”), the game rules, the productivity, defect, cost, time to market graphs, and the bubbles over the employees’ head. <i>Game score</i> : Students’ scores can only be seen at the end of the game. <i>Game Progress</i> : During the game the players can see information related to the results of their actions (e.g. employees’ mood and energy). <i>Play strategy</i> : Students can play the game immediately after reading the project goal, using a “trial and error” strategy (exploratory interaction), since the game does not direct them towards any other information. <i>Events</i> : Random events (de-motivator and motivator factors) can occur during the whole game. Independent events cause the action to begin automatically, without player interaction.

mathematical simulation model. The simulation model represents aspects of the world to be simulated and it is defined externally in an XML file. This model took into account the player’s performance in order to generate the final score. Monitoring the states is the main integration skill assessment mechanism. SPIAL monitors the time, the project metrics, process areas capability levels, and the development disciplines (requirement, design, development or test) of a software development project. These states are used for the calculation of the final score. The transition of the game states are triggered by player actions, random events, or autonomous events, i.e. without player interaction. Introducing unique and unexpected events promotes the attention and interest of the learner. We observed that all these assessments happen during the game play, they are embedded into the game environment (in-process as primary type). Finally, these assessments are more oriented to the learning of the players than to the certification of their skills.

5. SPIAL External Skill Assessment Mechanisms

External assessment was implemented in different ways, including an inspection method, pre and post tests and user-test. In the first evaluation, we used a Semiotic Inspection Method (SIM) from the Semiotic Engineering [de Souza et al. 2006]. In this method, an inspector carried out the assessment, assuming the role of a student. After conducting the SIM steps, a unified analysis was produced, highlighting the main communication breakdowns. According to the inspector, important aspects to understand the core behavior were missing, such as the reason why sometimes investments do not produce any improvement. We produced a new game version, with some modifications, in order to provide more guidance to players.

In the second evaluation, we carried out two experiments with undergraduate students of a Software Engineering course. In the first experiment, our aim was to gain a better understanding of SPIAL effectiveness as an educational tool. Specifically, we observed students’ understanding, remembering and application of Software Engineering concepts in the context of CMMI based SPI initiatives using SPIAL. In addition, we verified the adequacy of SPIAL in terms of its design, content, duration and student’s engagement. We also determined the strengths and weaknesses of SPIAL through the feedback of the students who played it. In total, 11 undergraduate Computer Science students participated in this pilot experiment. Each student answered a background and a

pre-test questionnaire, before playing SPIAL, and a post-test, after playing it. The pre-test and post-test were composed of 16 specific questions about SPI (with multiple choice and open questions), based on the CMMI, e.g. “List three metrics that can be used to monitor a Software Process Improvement initiative.” We have not identified a significant difference between the pre and post-test. Only a few students presented a gain in software process knowledge. Considering the engagement, on average, students found SPIAL quite enjoyable and they had fun during the game play. They also felt that the game duration was appropriate and it was relatively easy to play. They agreed about adopting SPIAL in a Software Engineering course as a complementary approach. The students considered the game content relevant to their learning. They felt that SPIAL reflects aspects of a real Software Process initiative. They observed that this game was sufficient when considering its purpose and it had a satisfactory play sequence. All students agreed that a traditional Software Engineering class with SPIAL will be better than without it. They moderately learned new concepts and practical application of an SPI program in an organization. They felt that SPIAL was more successful in reinforcing concepts taught in Software Engineering course than teaching new concepts.

In the second experiment, we applied a specific framework [Peixoto et al. 2014] in order to evaluate the game experience, adaptivity, learning experience and usability dimensions. The students answered specific questions for each dimension, with respect to the Understandability/Usability dimension an example of question is “how easy was the understanding of the game inputs and the outputs?” In total, 15 students participated in this experiment. The highest scored attribute was the learnability, i. e. the students understood how the game play works (Table 3). The attribute with smallest score was the learning goals, and this reflects the fact that players did not feel that they gained a great amount of new knowledge.

Table 3. Average Results for UGALCO’s Attributes. Extracted from Peixoto and colleagues [Peixoto et al. 2014].

<i>Dimension</i>	<i>Attribute</i>	<i>Average</i>
Game Experience	Challenge	0,47
	Competence	0,68
	Immersion	0,70
	Positive affect	0,65
Adaptivity	Cognitive and motivational aspects	0,60
Learning Experience	Learning goals	0,46
	Content appropriateness	0,63
Usability	Operability	0,45
	Understandability	0,63
	Learnability	0,77
	Attractiveness	0,47
	Satisfaction	0,62

Although the first two evaluations provided us with important data, the insight gained into an individual student’s play process was limited to questionnaires. Therefore, we conducted a simple user test, where two students were observed and then interviewed about their experience. The user test occurred in a one-to-one setting: one subject and one evaluator. As corroborating with the previous experiments, students reported that they reinforced SPI concepts playing SPIAL, but they were not sure about what new concepts were presented to them.

In these evaluations, we observed that the most frequently occurring outcomes

were the reinforcement of concepts, and motivational improvement. We were not able to trace the real learning effects considering these assessments. With the experiments, we could not statistically demonstrate the learning effectiveness of this game. A more comprehensive evaluation mechanism, using pre and post game playing questionnaires with specific questions, should be elaborated.

6. Discussion

In this section, we discuss the findings of the assessments conducted in SPIAL. We also debate the challenges that we faced and improvements that we can implement in future works.

6.1. In and out-assessments

The external evaluation of SPIAL did not consider internal actions of the game. For instance, the sequences of investments that the player has made in-process areas were not the target of any questions on the external evaluations. This information is important to understand the patterns of players' behavior in solving problems. This is also useful information when evaluating their behaviors when facing a problem and the impact on the learning process.

Therefore, improvements should be incorporated in SPIAL in order to provide information to support the integration of internal and external assessments. Otherwise, relevant information for a complete assessment could be lost. For instance, collecting and analyzing players' reactions to events or players' behaviors.

6.2. In- assessment and skill certification

The learning outcomes on a course are defined as skills, and students must be evaluated according to the proficiency level on these skills [Caballero-Hernández et al. 2017]. However, in SPIAL, the in-game assessments were applied mostly for guiding students during the learning process. They are not really used for evaluating the acquired skill after the players have experienced some situation. Therefore, in SPIAL, in-game assessments could be remodeled in order to better achieve their main goal.

6.3. Out-assessments

The assessments of learning effectiveness in Software Engineering simulation games employ a mixed-method approach, covering a few characteristics of the game environment [Peixoto et al. 2014]. In the literature, we found relatively few studies that have definitively assessed the effectiveness of the educational game [Connolly et al. 2012, Calderón and Ruiz 2015].

After the SPIAL evaluations, we were not able to affirm that students learned a considerable number of new concepts. The statistical tests did not show difference when we applied the SPI pre and post-questionnaires. We observed that making comprehensive evaluations of the learning effects is an important field of research, since until now most of Software Engineering simulation games only evaluated the students' perception of learning.

6.4. Single player assessment

Most of the researches are focused on assessing single player games. In a context with more than one player, the assessment difficulties and problems are escalated. So, assessment in an environment with multiple players is not a simple task. It is needed more investigation in order to obtain valuable results and improvements in this research area.

7. Conclusion

In this paper we used four assessment categories in order to evaluate the SPIAL assessment mechanisms. The embedded assessment mechanisms were identified with the evaluation of the game signs. External assessments were carried out in different ways, including an inspection method, pre and post tests and user-test. We observed that the SPIAL learning effectiveness was evaluated during the external assessments, using questionnaires. However, in our case, the statistical tests did not show difference. We detected other limitations in the assessment methods, for instance, the certification of the obtained skill is mostly implemented out of the game. The game is mainly used as a formative tool, but the Integrative aim is not suitably explored. We believed that SPIAL feedback does not improve player's autonomy capacity.

In general, improvements should be incorporated into the SPIAL environment in order to allow the integration between internal and external assessments. In addition, other assessment mechanisms should be integrated in the game environment, such as summative evaluations and integrative aim.

References

- Barros, M. d. O., Dantas, A. R., Veronese, G. O., and Werner, C. M. L. (2006). Model-driven game development: experience and model enhancements in software project management education. *Software Process: Improvement and Practice*, 11(4):411–421.
- Boyle, E. A., Hainey, T., Connolly, T. M., Gray, G., Earp, J., Ott, M., Lim, T., Ninaus, M., Ribeiro, C., and Pereira, J. (2016). An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games. *Computers & Education*, 94:178 – 192.
- Caballero-Hernández, J. A., Palomo-Duarte, M., and Dodero, J. M. (2017). Skill assessment in learning experiences based on serious games: A systematic mapping study. *Computers & Education*, 113:42 – 60.
- Calderón, A. and Ruiz, M. (2015). A systematic literature review on serious games evaluation: An application to software project management. *Computers & Education*, 87:396 – 422.
- Carrington, D., Baker, A., and van der Hoek, A. (2005). It's All in the Game: Teaching Software Process Concepts. In *Proceedings of the 35th Frontiers in Education Conference*, FIE'05, page F4G, Indianapolis, Indiana.
- Chaudy, Y., Connolly, T., and Hainey, T. (2013). Specification and Design of a Generalized Assessment Engine for GBL Applications. In *7th European Conference on Game Based Learning*, pages 105–114.
- CMMI (2010). CMMI ®for Development, Version 1.3. Technical Report CMU/SEI-2010-TR-033, Software Engineering Institute.

- Commission, E. (2012). Education and Training 2020 Work Programme. Technical report, Directorate-General for Education and Culture.
- Connolly, T. M., Boyle, E. A., MacArthur, E., Hainey, T., and Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education*, 59(2):661 – 686.
- Crisp, G. (2012). Integrative assessment: reframing assessment practice for current and future learning. *Assessment & Evaluation in Higher Education*, 37(1):33 – 43.
- de Souza, C. S. (2005). *The Semiotic Engineering of Human-Computer Interaction*. The MIT Press.
- de Souza, C. S., Leitão, C. F., Prates, R. O., and da Silva, E. J. (2006). The Semiotic Inspection Method. In *Proceedings of the VII Brazilian Symposium on Human Factors in Computing Systems, IHC '06*, pages 148–157, Natal, Brazil. ACM.
- Jain, A. and Boehm, B. (2006). SimVBSE: Developing a Game for Value-Based Software Engineering. In *Proceedings of the 19th Conference on Software Engineering Education and Training*, pages 103 –114, Oahu, Hawaii.
- Kupsch, D. C. C. (2012). *SPIAL: A tool for software process improvement training*. PhD thesis, Universidade Federal de Minas Gerais, Brazil.
- Mandl-Striegnitz, P. (2001). How to successfully use software project simulation for educating software project managers. In *Proceedings of the 31st Frontiers in Education Conference*, Reno, Nevada.
- Navarro, E. O. (2006). *SimSE: A Software Engineering Simulation Environment for Software Process Education*. PhD thesis, Donald Bren School of Information and Computer Sciences, University of California, Irvine.
- Peirce, C. S. (1992). *The Essential Peirce (Vols. I and II)*. Indiana University Press, Bloomington, Edited by Nathan Houser and Christian Kloesel edition.
- Peixoto, D. C. C., Possa, R. M., Resende, R. F., and Pádua, C. I. P. S. (2012). Challenges and Issues in the Development of a Software Engineering Simulation Game. In *Proceedings of the 42nd Frontiers in Education Conference, FIE'12*, Seattle, Washington.
- Peixoto, D. C. C., Resende, R. F., and Pádua, C. I. P. (2014). Evaluating Software Engineering Simulation Games: The UGALCO Framework. In *Proceedings of the 44th Frontiers in Education Conference, FIE'14*, Madrid, Spain.
- Petri, G. and von Wangenheim, C. G. (2017). How games for computing education are evaluated? A systematic literature review. *Computers & Education*, 107:68 – 90.
- Qian, M. and Clark, K. R. (2016). Game-based Learning and 21st century skills: A review of recent research. *Computers in Human Behavior*, 63:50 – 58.
- Vos, L. (2015). Simulation games in business and marketing education: How educators assess student learning from simulations. *The International Journal of Management Education*, 13(1):57 – 74.