Games for Teaching Computing in Higher Education – A Systematic Review

Paulo Eduardo Battistella, Christiane Gresse von Wangenheim

***Abstract*— Games are becoming popular for computing education, as they may increase learning effectiveness and engagement. In order to understand which kind of games exists for teaching computing in higher education, we performed a systematic review. We identified 107 games, mainly for teaching software engineering, programming fundamentals, networks, algorithms & complexity and security. Digital games are still predominant, although there can also be observed a trend to non-digital ones (e.g. board games). And, although, the games seem to approach well the respective knowledge areas, there seems to be a lack of their incorporation into the learning context. Future research should, therefore, focus on their design based on instructional and game theory to assure the intended learning outcomes.**

***Index Terms*— computing education, educational game, serious game, systematic literature review**

I. Introduction

INFORMATION Technology is a basic enabler in our lives today, impacting how we work, communicate and are entertained [1]. The demand for skilled computing professionals is continuously increasing as are the demands for more effective higher education. The computing discipline needs to attract quality students and to educate them to be capable and responsible professionals in the 21 century [2]. At the same time computing education has to be appealing in a way that motivates and challenges them. Yet, although the nature of computing and the generation of students has changed remarkably in the past years, most undergraduate computing courses are still taught in traditional ways that may not be adequate to keep up with modern concerns [3][4].

An approach taken to stem the declining interest and enrollment in computing courses and to reach students in an engaging way is game-based learning availing the passion of students to video games [5]. Game-based learning[1] deals with games that have defined learning outcomes [6][7]. They are designed in order to balance subject matter with gameplay. A game can be defined as "any contest (play) among adversaries (players) operating under, and educational (or serious) games are specifically designed to teach people about a certain subject, expand concepts, reinforce development, or assist them in drilling or learning a skill or seeking a change of attitude as they play [8]. Games are being considered a powerful instructional constraints (rules) for an objective (winning, victory or pay-off)" [6] method and are believed to result in a wide range of benefits, increasing learning effectiveness, interest, motivation and persistence [9][10][11][7]. They can promote "active learning" to achieve deep learning within acceptable teaching time and instructor load [12]. Furthermore, they can serve as an entertaining means for drill and practice as well as to illustrate dynamics or abstract principles [13]. And, due to their inherent characteristics, such as, competition, challenge and interaction, they can turn learning into an engaging experience having fun [14].

Therefore, game-based learning also seems to be a promising alternative to teach computing in higher education. Yet, there seems to be a lack of synthesized information on which games exist as well as on how to develop games to teach computing competencies, which may hinder a broader adoption of this instructional method in practice. And, although there exist literature reviews that aim at the (meta-)analysis of the impact of games for teaching (e.g., [15][16][17][18]), they typically cover either a broader scope on any kind of subject matter (e.g., [15] [16]) or are limited to a specific computing knowledge area, such as, software engineering ([12][19][20]) or programming fundamentals [21]. Another limitation is that several of the reviews focus on digital games only ([8][16][21] [20]).

Thus, in order to guide the adoption and/or creation of educational games for teaching computing, this article aims at reviewing and synthesizing information on existing games for teaching computing in higher education. Specifically, we address the following questions: (1) which games are used for computing education focusing on which knowledge areas? (2) to achieve which kind of learning goals? (3) which kind of games are used? and (4) how are these games designed and evaluated? We expect the results of our research to provide an

This work is supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico – www.cnpq.br) and CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - http://www.capes.gov.br), entities of the Brazilian government focused on scientific and technological development.

Paulo Eduardo Battistella is a Phd student at the Federal University of Santa Catarina, Florianópolis, Brazil (e-mail: paulo@incod.ufsc.br).

Christiane Gresse von Wangenheim is a professor at the Federal University of Santa Catarina, Florianópolis, Brazil (e-mail: gresse@inf.ufsc.br).

[1] Other kinds of usage of games for instruction such as learning by making games (e.g., teaching introductory computing concepts by having the students program new games) or gamification are beyond the focus of this article.

overview of existing games for instructors to facilitate the adoption of games in computing education as well as a source for creative ideas for new innovative games. The results of the review are also intended in an explorative way to provide a picture on open research issues in this area, guiding future research efforts in a systematic way.

The paper is structured as follows. In section 2, we present the methodology adopted for the systematic literature review. Results with respect to each of the research questions are presented in Section 3. Section 4 summarizes and discusses the results and conclusions are given in Section 5. Detailed information extracted from the encountered material is documented in the appendix.

II. METHOD: HOW THE GAMES HAVE BEEN SELECTED AND ANALYZED

This review was done to systematically synthesize information on games for teaching computing in higher education. We follow a general systematic literature review process [22]. Specifically, we address the following research questions: (1) which games are used for teaching which computing knowledge areas? (2) to achieve which kind of learning goals? (3) which kind of games are used? and (4) how are these games designed and evaluated?

Inclusion and exclusion criteria. We examined all English-language material (articles, manuals, websites, etc.) on games for teaching computing that was available on the Web since 2000. Our search covers the teaching of any kind of computing competencies (including knowledge, skills and attitude). Yet, games exclusively destined to train company specific products and not general computing knowledge have been excluded. We considered any kind of game with an instructional purpose, including digital and non-digital games. On the other side, we excluded any other kind of instructional method such as e.g., case studies, problem/project-based exercises or simulations. We also excluded any material reporting on how to use commercial games such as e.g., Scrabble, Tetris, for teaching. We focused on games used for learning through games, excluding any material on learning by making games, where students develop games and in that way learn (e.g., by programming a game as an exercise in introductory computing courses) or gamification approaches. The focus of our study is on higher education (especially on the undergraduate level), excluding any games not applicable to this context. We also excluded material describing only proposals or conceptual models of games that have not (yet) been developed. We analyzed all games for which at least an English-language description was available on the Web, even if the game itself is not available in English.

In terms of quality, the available material on the game had to be sufficient to understand the game and its characteristics. We also excluded games which were cited only, but for which could not be encountered a sufficient description and/or a game version that could be played in order to analyze the game.

Data sources and search strategy. We searched the Web via Google in order to obtain a broad overview, not only considering games published through academic articles but also commercial games, etc. We searched for “game” and “education” (using also several related terms, such as, learning, teaching) in combination with terms representing core computing knowledge areas. In accordance to the length restrictions of Google search strings to 32 words, we divided our search in several strings as shown in Table I.

TABLE I. SEARCH STRINGS

game (education OR educational OR learning OR teaching OR training OR "serious game") (computing OR "computer science")
game (education OR educational OR learning OR teaching OR training OR "serious game") ("information systems" OR "Information Technology" OR IT)
game (education OR educational OR learning OR teaching OR training OR "serious game") "artificial intelligence"
game (education OR educational OR learning OR teaching OR training OR "serious game") (algorithms OR “data structures”)
game (education OR educational OR learning OR teaching OR training OR "serious game") “programming languages”
game (education OR educational OR learning OR teaching OR training OR "serious game") ("computer network” OR network)
game (education OR educational OR learning OR teaching OR training OR "serious game") compilers
game (education OR educational OR learning OR teaching OR training OR "serious game") “computer architecture"
game (education OR educational OR learning OR teaching OR training OR "serious game") “computer graphics”
game (education OR educational OR learning OR teaching OR training OR "serious game") database
game (education OR educational OR learning OR teaching OR training OR "serious game") “distributed systems”
game (education OR educational OR learning OR teaching OR training OR "serious game") "digital logic"
game (education OR educational OR learning OR teaching OR training OR "serious game") “embedded system”
game (education OR educational OR learning OR teaching OR training OR "serious game") “human-computer interaction”
game (education OR educational OR learning OR teaching OR training OR "serious game") “operating systems”
game (education OR educational OR learning OR teaching OR training OR "serious game") programming
game (education OR educational OR learning OR teaching OR training OR "serious game") security
game (education OR educational OR learning OR teaching OR training OR "serious game") (“software requirements” OR “software architecture” OR "software design" OR “software engineering” OR “software process” OR “software quality” OR “software verification” OR “software validation” OR “software testing” OR “software maintenance”)
game (education OR educational OR learning OR teaching OR training OR "serious game") “system administration”
game (education OR educational OR learning OR teaching OR training OR "serious game") ”scientific computing”
game (education OR educational OR learning OR teaching OR training OR "serious game) “numerical methods”
game (education OR educational OR learning OR teaching OR training OR "serious game") ethics

Study Identification and Selection. Initial searches in December 2012 returned a total of 1.506.000.000 results. We analyzed the first 300 results for each of the searches ordered by decreasing relevancy observing a rapid reduction of relevancy after the first 100 results. In the first stage, we quickly reviewed headings and short summaries with regard to the inclusion criteria and excluded irrelevant or duplicated material, selecting 194 games. In order to further analyze their relevance, we either analyzed the encountered material (academic papers, manuals,

etc.) and/or played the games. Many of the material encountered describe simulations for teaching computing, especially, e.g., in the area of computer architecture. Yet, due to our focus we excluded any kind of pure simulation that does not present game characteristics.

In addition to our inclusion/exclusion criteria, we also superficially assessed the quality of the identified material, considering only games for which a sufficient description or access to the game (or a demo version) was available that allowed the extraction of sufficient information with regard to our research. If available, we also took into consideration secondary literature for which we searched by using the game names. Using this criterion, 107 games were identified as relevant to the review (see Table II, at the end of the article).

Data extraction and checking. For each identified game that met the inclusion and quality criteria, we extracted information characterizing and classifying the games in terms of instructional and game aspects with respect to the research questions. For data extraction, the material has been analyzed thoroughly and, if available, games have been played. Data extraction, especially, regarding information on the instructional aspects was hindered in many cases by the way in which the games were described. Most papers lack sufficient detail about instructional aspects such as the learning objectives. In addition, most papers do not provide any information on the instructional game design. In case of digital games, information on the game development process is generally limited to the software implementation. If available, also descriptions of the game's evaluation lack details such as a clear definition of the adopted the research design, and do not report the studies in alignment with a research evaluation framework or guidelines. Thus, some information was inferred by the authors based on the information reported. The extracted information is presented in detail in Table II and the Appendix.

Data analysis. Due to explorative nature of our research and the available data, we performed a qualitative analysis of the extracted information with respect to the research questions. The results are presented in Section 3.

III. RESULTS

In this section, we present the results of the systematic review providing an overview on the state-of the-art of games for teaching computing in higher education.

(1) Which games are used focusing on which computing knowledge areas?

In general, we encountered a considerable number of 107 games (Table II). This demonstrates that there exists a trend to game-based learning also in computing education.

A breakdown of the available games by computing knowledge areas reveals that the majority of the games focuses on teaching Software Engineering (SE) (Figure 1). This may be due to the specific requirements in SE education, where students have to be given the opportunity to apply SE concepts to practice situations [23]. Yet, as often there is not sufficient time in SE courses to provide students with a solid understanding and practical experience, alternative instructional methods including games may be an important alternative. This is also reflected by the fact that many of the SE games are simulation games, in which the player assumes the role of a project manager and then has to plan, execute and monitor & control a software project (e.g., [24][25][26][27][28][29][30][31]).

Other knowledge areas for which a considerable amount of games exist are programming fundamentals, computer networks, algorithms & complexity and security. Using games to teach programming fundamentals can help especially beginners to visualize the effect of their programming reinforcing the learning effect. For example, Wu's Castle [32], requires the player to instantiate program variables for FOR loops and, then, immediately visualizes the execution of the program by building the programmed quantity of snowmen.

Similar to SE games, games for teaching computer networks are also typically simulations, in which the job of the player is to identify problems in subnets and fix them so the network works properly [33][34][35]. In this knowledge area, games also help to enable the simulation of realistic scenarios, teaching required competencies within typical class restrictions. This can also be observed in relation to games used for teaching security [36].

Games for learning algorithms & complexity are strongly focused on having the students execute different algorithms and to experience complexity issues – coupled with a competition between players [37][38][39]. Thus, they represent a way active to involve the learners, in an entertaining way, in order to contribute with the understanding and memorization of the knowledge to be learned.

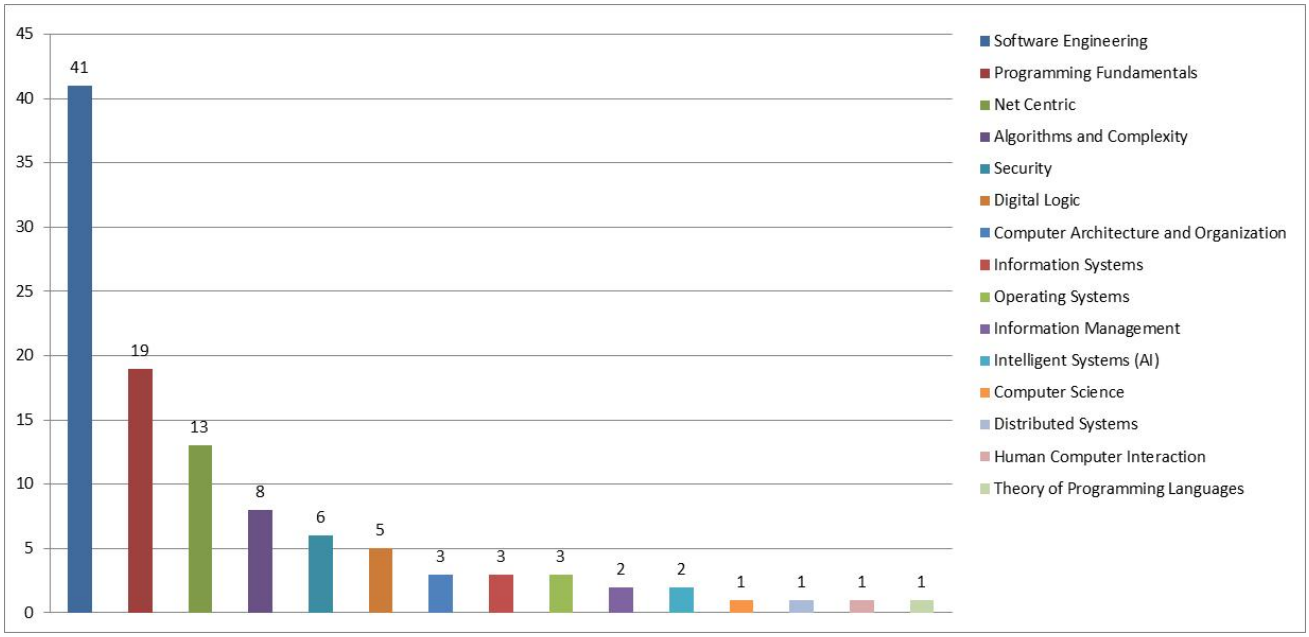


Fig. 1. Number of games per computing knowledge area.

A certain number of games have also been encountered for learning digital logic (e.g., [40][41][42]). This seems to be motivated by the need to demonstrate the functioning of the concepts, as besides the games found, there also exist a lot of simulators without a game component for this knowledge area.

For several other knowledge areas, including, information management, human computer interaction, theory of programming languages, computer graphics etc., very few or no games at all have been encountered. This shows that, although, a considerable number of educational games for teaching computing competencies exist, most of these games concentrate on teaching a small number of specific knowledge areas.

(2) Games are used to achieve which kind of learning goals?

On the undergraduate level, it is expected that students acquire computing competence, mainly, on the cognitive levels of knowledge, comprehension, and application in accordance to Bloom's taxonomy of learning objectives [43][44][45]. Yet, besides computing knowledge, students also need to develop skills, such as, communication, teamwork, and management in relation to the discipline [2]. Such learning objectives can be classified by re-interpreting levels of psycho-motoric learning objectives focusing on the change and/or development in behavior or abilities [46]. Furthermore, computing programs also need to encourage students to develop a mature attitude toward practicing computing in a professional, responsible, and ethical manner. Such learning objectives are typically targeted to the awareness and growth in attitudes, emotion, and feelings [44]. In this regard, levels of learning objectives for each of the domains can be classified as illustrated in Figure 2.

Fig.2 Levels of learning objectives

By inferring from the game descriptions, we can observe that the great majority of the games aims at learning on the cognitive level (figure 3). Only a very small number of games aims at teaching skills (such as, communication or teamwork [47][38]) or a change of attitude (e.g., the recognition of the difficulty and importance of team management [48]).

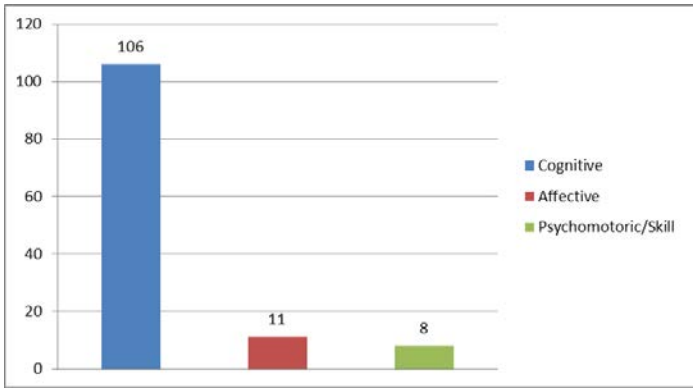


Fig. 3. Number of games per type of learning objective (more than one classification possible).

Considering each level of learning objective separately (Figure 4), we can observe that most games aim at lower cognitive levels in accordance to learning expectations on the undergraduate level. In this context, games seem to be much

more used to reinforce knowledge previously taught using different instructional methods than to teach new knowledge. Learning objectives related to skills focus exclusively on the mechanism level aiming at the learning of certain ability, such as, communication, teamwork, decision making or problem-solving. Affective learning objectives aim mainly at the recognition of certain values and/or creation of value-systems.

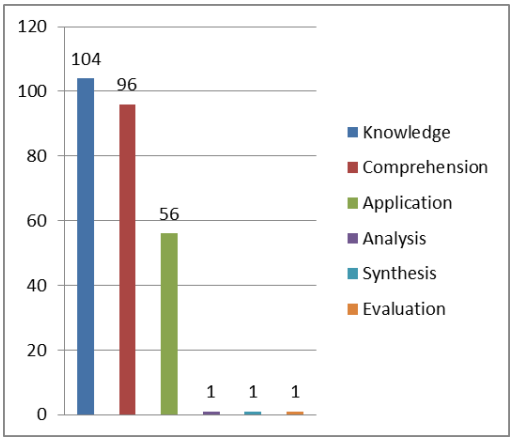


Fig. 4a. Number of games per level of cognitive learning objectives.

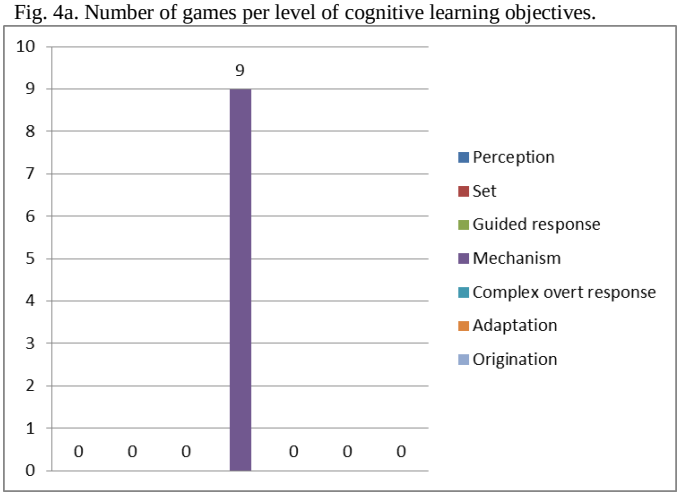


Fig. 4b. Number of games per level of skill learning objectives.

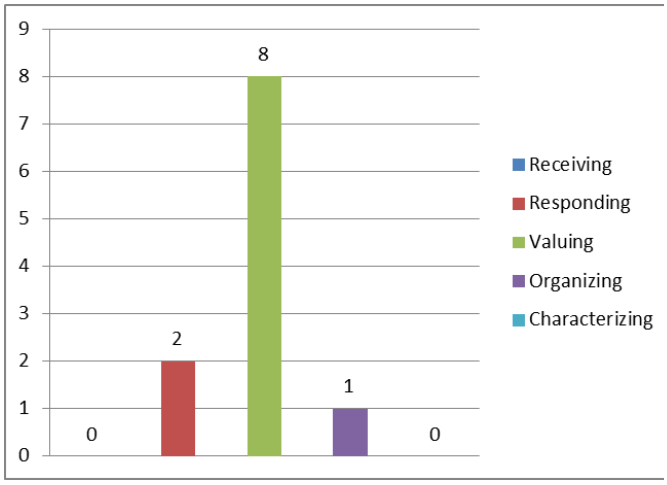


Fig. 4c. Number of games per level of affective learning objectives.

Most games are intended to be played once, especially, non-digital games. An exception is Hard Choices [49], which is expected to be played at least twice in order to gain understanding. But, also only a few digital games (e.g., [50]) seem to be designed to accompany the learning process on different levels with increasing difficulty.

Another issue regarding the instructional component of these games is the question on how feedback is given to the learners on their performance. Yet, again most material does not provide detailed information on this issue. And, although, several digital games typically provide a game score and/or error log afterwards, they seem to lack more constructive guidance on what has been done wrong and how to correct this and/or where to find further information. In case of non-digital games, the feedback is mostly given through debriefing sessions and/or through the instructor.

(3) Which kinds of games are used?

As also typical in other areas, the majority of the games we encountered for teaching computing are digital games (68 games). Yet, there is also a considerable number of non-digital (40 games), which are growing in popularity, a trend that can also been observed in entertainment recently [51]. It seems that, especially, the fact that they are played as a group, sharing a physical representation, brings several benefits by increasing social interaction, competition and, thus, engagement. As many digital games are single-player, they do not invoke these reactions so strongly. This also provides an often overlooked benefit in using games: the opportunity to share experiences and exchange knowledge. Another major advantage is that non-digital games have a very low development threshold, while on the other hand digital games require a considerable development effort [52]. Furthermore, they are also much easier to be tailored or enhanced to the specific needs of an instructional unit by changing the rules or inserting extra exercises that expand or modify the game experience.

Analyzing the game platform in more detail, we can observe that among digital games PC games are predominant (Figure 5a). And, although, there exist a lot of standalone games, the trend is clearly towards online games. So far, few games make usage of social network elements, such as, Second Life [47] or exhibiting game scores via Facebook [53]. Console games are basically not used for teaching computing – with the exception of BINX [41]. Until now, also mobile games are basically inexistent with only one exception [54]. Regarding non-digital games, a large number are either games that require only paper & pencils to be played or board games (Figure 5b). Less common are card or prop games, that use other objects, such as, Lego [55] or oranges [56] for gameplay.

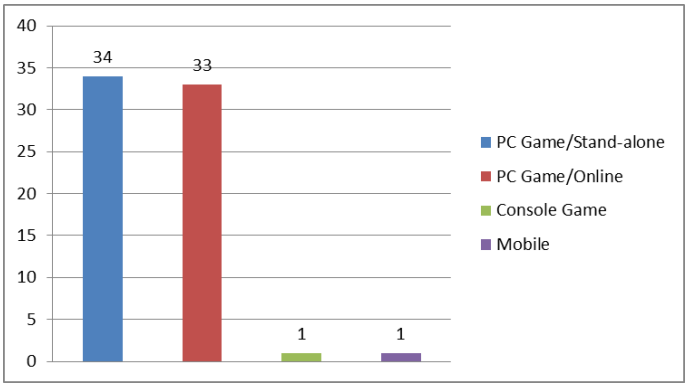


Fig. 5a. Number of digital games per platform.

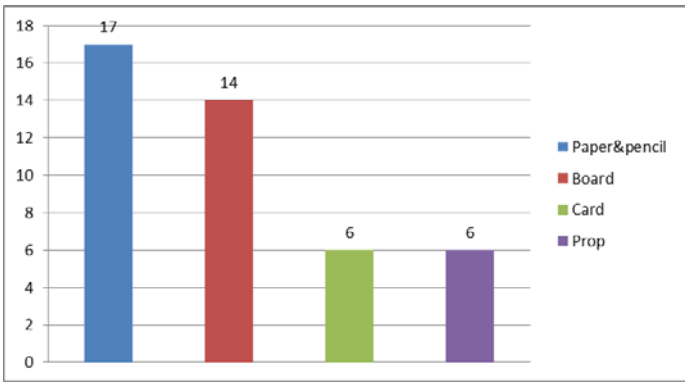


Fig. 5b. Number of non-digital games per platform.

In terms of genre, we can observe that most games are simulations (Figure 6). This reflects a general tendency in using simulation games for teaching to replicate real-world contexts as they challenge students to analyze available information and make critical decisions based on theoretical and practical knowledge. Simulations are accepted as a complementary mode for traditional teaching methods by stimulating and motivating learning through providing a realistic environment for the students to practice the competencies to be learned. Furthermore, as such games represent a combination of simulations that allow to visualize complex concepts with game components, they allow an active learning approach provoking experimental learning while keeping students engaged. They can involve students in a more enjoyable simulated experience of the real world. Many of the non-digital games are simulations (e.g., [57][55][58]). In these games, players typically have to create objects, such as, paper hats or LEGO houses by applying relevant concepts from project management or requirement analysis.

We also encountered several puzzle games, mainly, for learning programming, where the players have to program robots or other objects to solve problems such as to navigate through a maze (e.g., [59][60][61]). There exist also a certain amount of games where the player participates in an interactive adventure and has to solve challenges and quests in order to win, such as, decoding messages. Examples are hACME [62] or Secret Ninja Testing [119]. We also encountered some Role-Playing Games, where the player controls the actions of a protagonist that lives immersed in a fictional world, as, e.g., Saving Princess Sera [32] or JV2M [63]. Other genres used to teach computing include action, roll-and-move games, strategy, quiz, guessing and racing games.

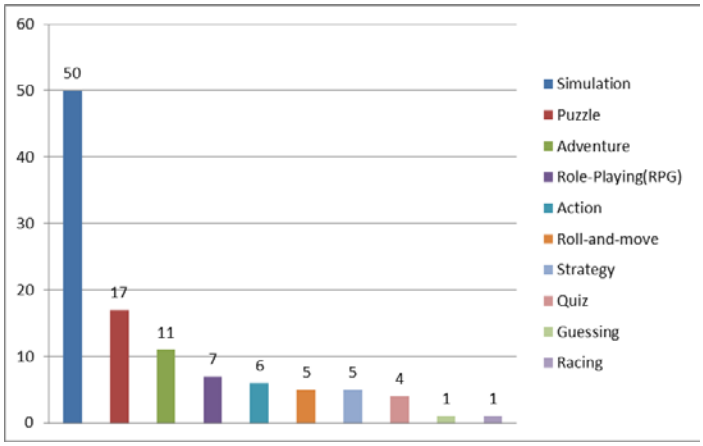


Fig. 6. Distribution of games per genre.

In terms of interaction mode, we can observe that the majority of digital games are single-player. Yet, this emphasis on single-player games means that social play and interaction while playing is separated from the "game". So far, only few digital games have been encountered that as multi-player games provide the opportunity to form cooperations, competition or rivalry and, thus, contribute to engagement. On the other hand, the social interaction provided by non-digital games is one of their outstanding strengths, as they are usually played in groups. When playing non-digital games, there commonly is an exciting atmosphere of communication and competition that often besides the intended learning objectives also encourages verbal communication abilities, social skills and traits of patience and persistence. Playing non-digital games has even been recognized to help students to build self-confidence and self-esteem while interacting with others in a positive way [64][31].

Another practical aspect is the time it takes to play the games. Although, this information is rarely given in the game material, most of these games seem to be either designed to fit into the typical duration of a class (about 90 min) or to be played very rapidly (about 10 min). Games for teaching computing also seem to be more designed for a selective application at one specific point in the instructional context. Few games are designed to accompany a whole instructional unit, as, e.g., Z-Buffer [65], where the availability of levels depends on the player's progress in the game in order to ensure a pedagogical order of introduced topics that fits the pace of the instructional unit.

(4) How are these games designed and evaluated?

The design of instructional games is a research area that may be conceptualized as the intersection of learning and, game theory, and subject matter expertise. Yet, it is surprising that the majority of the material describing the games does not include any kind of description on how the games have been developed. Especially, when regarding the instructional objective(s) of

these games, a systematic description on how these games are aligned with instructional theory and design is expected, yet not to be found in most cases. Information on the instructional context, including course, target audience or instructional strategies is rarely provided. Few exceptions include the Anti-Phishing Phil game for which the complete development process is described [66]. Other examples approaching instructional design include [67][68] [57] citing the usage of the ADDIE [69] or the ISD model [70]. Two game descriptions also indicate the usage of concept maps for instructional design [65][71]. Little more information is generally provided in the case of digital games on the software development (e.g., [72][41][63]) either describing details of the implementation or indicating game platforms and technologies used [73][74][75][32][40][76]. Hardly any game material explains in detail game design elements, such as, competition, reward mechanism, interaction mode, or narrative.

On the other hand there seems a positive trend to evaluate such games, although often through a non-experimental approach only (Figure 7). A distinction here is the series of replications of different kind of studies (including experiments) aiming at a multi-angled evaluation of the SimSE game [77][29], besides a few other (quasi)-experimental evaluations.

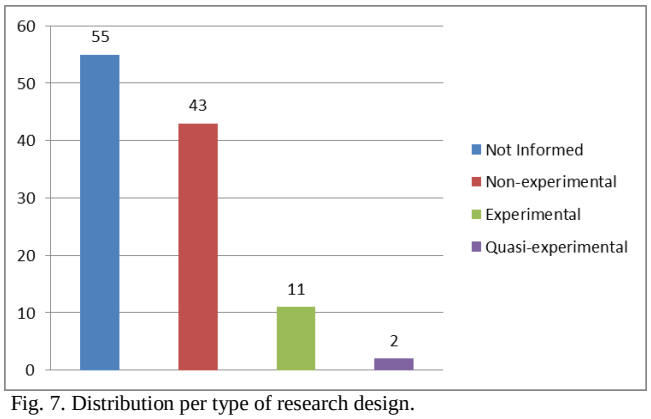


Fig. 7. Distribution per type of research design.

Yet, the most common research design are case studies using a one-shot post-test only design, where, typically, the game is used and afterwards subjective information is collected via questionnaires from the learners. Most studies aim at the evaluation of the reaction of the learners, by capturing their subjective perception. Fewer studies evaluate the games on the learning level actually comparing pre- and posttest results from before and after the gameplay. No longitudinal studies to analyze the transfer of the learned competency into the professional context have been encountered.

The evaluation purpose of most studies is to understand learning effectiveness and enjoyability as well as to identify strengths and weaknesses of the game. Only very few studies systematically compare the learning effectiveness of the games to other instructional methods [76]. Yet, based on subjective feedback, many studies report that learners evaluate the games very positively and prefer them in comparison to traditional instructional methods.

IV. Discussion

The general conclusion from the review is that there is a trend to teach computing with games in higher education as there exists a considerable amount of games. Various findings can be summarized:

- Games are mainly used for teaching software engineering, programming fundamentals, computer networks, algorithms & complexity and security.
- The majority are digital game, principally PC games, yet with a considerable trend also to non-digital games (paper & pencil, board, etc.).
- Simulation games, that allow to practice competencies in a realistic environment while keeping students engaged, are predominant.
- Most games aim at learning objectives on lower cognitive levels and are mostly used to review and reinforce knowledge taught beforehand using different instructional methods.
- These games seem to lack foundation in instructional and/or game theory, which may reduce their learning effectiveness and reduce the benefits of game elements such as competition, interaction and enjoyment.
- The common absence of evaluations and/or predominant adoption of non-experimental approaches makes it difficult to quantify and synthesize the impact of games in general.

Thus, there exist a clear trend and interest in using games for teaching computing and positive feedback from the students indicates that it can be a promising approach to increase learning effectiveness and to engage and motivate students in a fun way.

Yet, as many games currently seem to be developed mostly by subject matter experts themselves in an ad-hoc way, the need to systematize the development and their systematic foundation in alignment with instructional and game theory becomes obvious. In general, neither generic instructional design approaches (such as, ADDIE or ISD) seem to be applied nor specific frameworks that have been developed for the instructional design of games (such as, the ELEKTRA methodology [78][79]). Furthermore, there exists also the need for a larger number of experimental studies to increase the validity of the results. Hence, there seems to be the need for more multidisciplinary research bringing together subject matter experts with instructional and game designers in order to systematically explore and evolve this area.

A. Threats to validity

One of the main threats to the validity of this systematic literature review is incompleteness in relation to bias in the selection of games. To decrease the risk of incomplete search strings, we have included terms describing the relevant knowledge areas separately and several synonyms and related terms. A known set of games was used to calibrate the initial search terms. Taking into consideration that various games have not been published through academic articles, we tried to minimize any publication bias by not limiting our search to academic databases, but performing a broad search via Google.

We also considered, therefore, any kind of description or documentation. Despite this it is not possible to guarantee that all games relevant to the topic under consideration have been encountered.

In order to increase validity in relation to the selection of games, the selection has been done by both co-authors together in conformance with the defined inclusion and exclusion criteria and until consensus was achieved.

Data extraction and classification has also been done by both authors together and the resulting data has been checked and agreed upon. However, due to the common lack of information with respect to instructional and game aspects, we had to interpret and/or infer information in many cases, e.g., on the types of learning objectives, game genre or kind of evaluation study due to the lack of details. This might have resulted in a misclassification. However, in order to reduce this risk, taxonomies for the classifications have explicitly defined beforehand and both authors discussed the game classifications until achieving consensus.

V. Conclusions

The general conclusion from this review is that there is a trend to teach computing with games in higher education. In total, we encountered 107 games, mainly for teaching software engineering, programming fundamentals, computer networks, algorithms & complexity and security. The majority of these are digital games, mainly PC games, yet with a considerable trend also to non-digital games (paper & pencil, board, etc.).

However, there seems to be a certain lack of systematic development of these games especially with respect to instructional and/or game theory, most probably due to the fact that many of these games are developed by the subject matter experts themselves. As a consequence this may jeopardize the benefits of this kind of instructional method in terms of learning effectiveness and student's immersion and fun – not exploiting fully the inherent advantages of games. And, although a considerable number of games has been evaluated by some kind of empirical study, there seems to be missing broader and more long-term studies on the effects of instructional games also in comparison to other instructional methods.

Based on these findings, our recommendations are that using games as instructional method should be well founded and incorporated into the learning context by identifying the specific learning needs and designing them properly based on learning theory and instructional design models in order to enable the accomplishment of the intended learning outcome. It is also important to integrate the games within the learning context as a whole and, for example, include debriefing activities to complete a learning cycle. On the other hand, instructional games should also be designed in alignment with game theory in order to create engaging and motivating games. Consequently, there seems to be the need for more multidisciplinary research bringing together subject matter experts with instructional and game designers in order to systematically explore and evolve this area.

References

[1] M.K. Pratt, "10 hot IT skills for 2013", *Computerworld*, 2012. Available: http://www.computerworld.com/s/article/9231486/10_.

[2] ACM/IEEE-CS. *The Joint Task Force for Computing Curricula.* Computing Curricula 2005. *ACM/IEEE.* 2005.

[3] P. Parsons, "Preparing computer science graduates for the 21st Century", *Teaching Innovation Projects.* 2011.

[4] T. Barnes, E. Powell, A. Chaffin and H. Lipford, "Game2Learn: improving the motivation of CS1 students", *Proc. of the 3rd Int. Conf. on Game Development in Computer Science Education*, Miami, FL. 2008.

[5] B.B. Morrison and J.A. Preston, "Engagement : Gaming throughout the Curriculum", *Proc. of the 40th ACM Technical Symposium on Computer Science Education*, Chattanooga, TN. 2009.

[6] C.C. Abt, "Serious games", *University Press of America*. 2002.

[7] M. Prensky, "Digital game-based learning", McGraw-Hill. 2001.

[8] J. Dempsey, K. Rasmussen and B. Lucassen, "The instructional gaming literature: Implications and 99 sources", *Technical Report 96-1*, College of Education, University of South Alabama, AL. 1996.

[9] C.J. Kirriemuir and A. McFarlane, "Literature Review in Games and Learning Literature Review in Games and Learning", *Context*, 3(2), 208–213. 2004.

[10] A. Mitchell and C. Savill-smith, *The use of computer and video games for learning: a review of the literature*, Learning and Skills Development Agency, London, UK. 2004.

[11] S. de Freitas, "Using games and simulations for supporting learning", *Learning, Media and Technology, Digital Games and Learning*, 31(4), 2006, pp. 343–358.

[12] C.G. v. Wangenheim and F. Shull, "To Game or Not to Game?", *IEEE Software*, 26(2), 2009. p. 92–94.

[13] R.T. Hays and M.J. Singer, "Simulation fidelity in training system design: bridging the gap between reality and training", *Recent Research in Psychology*, Springer. 1988.

[14] Y.B. Kafai, "The educational potential of electronic games: from games-to-teach to games-tolearn", *Conference on playing by the rules: the cultural policy challenges of video games.* Chicago, IL. 2001.

[15] T.M. Connolly et al., "A systematic literature review of empirical evidence on computer games and serious games", *Computers & Education*, 59(2), 2012, pp. 661–686.

[16] F. Ke, "A qualitative meta-analysis of computer games as learning tools", In *Handbook of Research on Effective Electronic Gaming in Education*, IGIGlobal, USA. 2009.

[17] R.T. Hays, "The effectiveness of instructional games: a literature review and discussion", *Technical report 2005-004, Naval Air Warfare Center Training Systems Division*, Orlando, FL. 2005.

[18] J.F. Vogel et al., "Computer gaming and interactive simulations for learning: a meta-analysis", *Journal of Educational Computing Research*, 34(3), 2006, pp. 229–243.

[19] C. Caulfield, J. Xia, D. Veal and S.P. Maj, "A systematic survey of games used for Software Engineering education", *Modern Applied Science*, 5 (6), 2011a, pp. 28–43.

[20] T.M. Connolly, M. Stansfield and T. Hainey, "An application of games-based learning within software engineering", *British Journal of Educational Technology*, 38(3), 2007, pp. 416–428.

[21] F.W.B. Li and C. Watson, "Game-based concept visualization for learning programming", *Proc. of the 3rd Int. Workshop on Multimedia Technologies for Distance Learning.* Scottsdale, AZ. 2011.

[22] B. Kitchenham, "Procedures for performing systematic reviews", *Tech. Report TR/SE-0401*, Keele University, UK. 2004.

[23] A. Gkritsi, *Scrum Game: an agile software management game*. Master Thesis, University of Southampton, UK. 2011.

[24] A. Baker, E.O. Navarro and A. Van Der Hoek, A. "Problems and Programmers: an educational software engineering card game", *Proc. of the 25th Int.Conf.on Software Engineering*, Portland, OR. 2003.

[25] C. Caulfield, D. Veal and S.P. Maj, "Teaching Software Engineering Project Management – a novel approach for Software Engineering programs", *Modern Applied Science*, 5(5), 2011b, pp. 28–43.

[26] T. Hainey et al., "Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level", *Computers and Education*, 56, 2011, pp. 21–35.

[27] A. Martin, "The design and evolution of a simulation/game for teaching information systems development", *Simulation e Gaming*, 31(4), 2000, pp. 445–463.

[28] E. S.S. Monsalve, V.M.B. Werneck and J.C.S.D.P Leite, "Teaching software engineering with SimulES-W", *Proc. of the 24th Conference on Software Engineering Education and Training*, Honolulu, HI. 2011.

[29] E.O. Navarro, *SimSE: a software engineering simulation environment for software process education*. Doctoral Dissertation, University of California, Irvine, CA. 2006.

[30] Sharkworld. "A project management game", 2008. Available: http://www.sharkworldgame.com/.

[31] C.G. von Wangenheim, R. Savi and A.F. Borgatto, "DELIVER! – An educational game for teaching Earned Value Management in computing courses", *Information and Software Technology*, 54(3), 2012, pp. 286–298.

[32] M. Eagle and T. Barnes, "Experimental evaluation of an educational game for improved learning in introductory computing", *Proc.of the 40th Technical Symposium on Computer Science Education*, Chattanooga, TN. 2009.

[33] CyberCIEGE, 2011. cisr.nps.edu/cyberciege/.

[34] Secure MyPlanNet, 2010. "The Cisco Learning Network", Available: https://learningnetwork.cisco.com/docs/DOC-7635.

[35] Secure Volunteer Game, 2010. Available: https://learningnetwork.cisco.com/docs/DOC-7559.

[36] C. Jordan et al., "CounterMeasures: a game for teaching computer security", *Proc. of the 10th Annual Workshop on Network and Systems Support for Games*, Ottawa, Canada. 2011.

[37] Battleships "Computer Science Unplugged", 2002. Available: http://csunplugged.org/searching-algorithms.

[38] L. Hakulinen, "Card games for teaching data structures and algorithms", *Proc. of the 11th Koli Calling Int. Conf. on Computing Education Research*, Koli National Park, Finland. 2011.

[39] Sorting Algorithms "Digital Game Based Learning", 2012. Available: http://sortingalgorithmsgame.webs.com/.

[40] M. Harper, J. Miller and Y. Shen, "Digi Island: a serious game for teaching and learning digital circuit optimization", *NASA*, Langley Research Center, USA, 2011, pp. 869–889.

[41] R. Morsi and C. Richards, "BINX: an XNA/XBox 360 educational game for electrical and computer engineers", *Consumer Electronics Times*, 1(3), 2012, pp. 33–42.

[42] Y. Tang, S. Shetty and X. Chen, "Educational effectiveness of virtual reality games promoting metacognition and problema-solving", Proc. Of the *119th Annual Conf. of the American Society for Engineering Education*, San Antonio, TX. 2012.

[43] B.S. Bloom, "Taxonomy of educational objectives: the classification of educational goals", *Handbook I, Cognitive Domain*. 1956.

[44] D.R. Krathwohl, B.S. Bloom and B.B. Masia, "Taxonomy of Educational Objectives, the Classification of Educational Goals", *Handbook II: Affective Domain*. New York, NY. 1973.

[45] ACM/IEEE-SE. *The Joint Task Force on Computing Curricula IEEE/ACM*. Software engineering 2004: curriculum guidelines for undergraduate degree programs in software engineering. 2004.

[46] E.J. Simpson, "The Classification of Educational Objetives, Psychomotor domain", *Gryphon House*. 1972.

[47] E. Ye, L. Chang and J.A. Polack-Wahl, "Enhancing software engineering education using teaching aids in 3-D online virtual worlds", *Proc. of 37th Frontiers in Education Conference* Milwaukee, WI. 2007.

[48] O.P. Carvalho, "Educational game to support teaching interpersonal skills for software project management", *Project Thesis*, Information Systems, Federal University of Santa Catarina, Brazil. 2012.

[49] N. Brown, P. Kruchten, E. Lim, R. Nord and I. Ozkaya, "The Hard Choices Game Explained. White paper", *Software Engineering Institute/Carnegie Mellon University, Pennsylvania, PA*. 2010.

[50] O. Zwikael, and A. Gonen, "Project execution game (PEG): training towards managing unexpected events", *Journal of European Industrial Training*, 31(6), 2007, pp. 495–512.

[51] K. Core, "Dice and digital - rehabilitating the board game geek", *BBC News*. 2013. Available: http://www.bbc.co.uk/news/business–21615083.

[52] E. Kain, "Are board games better than video games?", *Forbes Business.* 2012. Available: http://www.forbes.com/sites/erikkain/2012/04/19/ar.

[53] B. Leong, Z.H. Koh and A. Razeen, "Teaching introductory programming as an online game", Department of Computer Science, National University of Singapore. 2011.

[54] CargoBot, 2012. Available: http://twolivesleft.com/CargoBot.

[55] Scrum LEGO City, 2009. Available: http://www.agile42.com/en/training/scrum-lego-city/.

[56] Orange Game. "Computer Science Unplugged: The orange game", 2002. Available: http://csunplugged.org/routing–and–deadlock.

[57] C.G. von Wangenheim, R. Savi and A.F. Borgatto, "SCRUMIA - An Educational Game for Teaching SCRUM in Computing Courses", *Journal of Systems and Software*. 2013. Available: http://www.gqs.ufsc.br/applying-scrum-welcome-to-s.

[58] C.M. Zapata, "Teaching software development by means of a classroom game: The software development game", *Developments in Business Simulation and Experiential Learning*, 36, 2009, pp. 156–165.

[59] Robot Trouble, "Djco-robot-trouble: XNA serious game teaching procedural programming", 2011. Available: https://code.google.com/p/djco-robot-trouble/.

[60] Robozzle. "RoboZZle a social puzzle game", 2011. Available: http://robozzle.com/.

[61] Treasure Hunt. TreasureHunt. 2011. Available: https://www.nonamesite.com/web/cs-stem/treasure-hunter.

[62] O. Nerbraten and L. Rostad, "hACMEgame: A Tool for Teaching Software Security", *Proc. of the Int. Conf. on Availability, Reliability and Security*, Regensburg, Germany. 2009.

[63] M.A. Gómez-Martín et al., "Not yet another visualization tool: learning compilers for fun", *Proc. of the 8th Int. Symposium on Computers in Education*, León, Spain. 2006.

[64] S. Itzhaki, and P. Matthews, "For Homeschoolers e Parents: Shyness in Children", 2013. Available: http://g4ed.com/index.php/for-homeschoolers/427-shyness-in-children.

[65] K. Kuk et al., "Use game based interactive multimedia modules to learning basic concepts on courses for computing science", *Electrical Review*, 2012, pp. 150–153.

[66] S. Sheng et al., "Anti-Phishing Phil : the design and evaluation of a game that teaches people not to fall for phish", *Proc. of the 3rd Symposium On Usable Privacy and Security*, Pittsburgh, PA. 2007.

[67] J. Melero et al., "Considerations for the design of mini-games integrating hints for puzzle solving ICT-related concepts", *Proc. of the 12th Int. Conf. on Advanced Learning Technologies*, Rome, Italy. 2012.

[68] M. Thiry et al., "Empirical study upon software testing learning with support from educational game", *Proc. of the 23rd Int.Conf.on Software Engineering and Knowledge Engineering*, Miami Beach, USA. 2011.

[69] M. Molenda, J.A. Pershing and C.M. Reigeluth, "Designing instructional systems", *In Craig RL ed., The ASTD training and development handbook.* New York, NY, 1996, pp. 266–293.

[70] Dick, W. and Carey, L. "The systematic design of instruction", *4th ed.* New York, Harper Collin. 1996.

[71] R.O. Chaves et al., "DESIGMPS: A game to support the teaching models of quality of software processes", *Proc. of the 17th Workshop on Informatics for Education*, Aracaju, Brazil. 2010. Available: http://www.treinacom.ufpa.br/index.php?option=com_content&view=article&id=6:desigmps

[72] T. Do and J. Lee, "A multiple-level 3D-LEGO game in augmented reality for improving spatial ability", *Proc. of the 13th Int. Conference on Human-Computer Interaction.* San Diego, CA. 2009.

[73] L.A. Belfore-II, A.B. Adcock and G.S. Watson, "Introducing digital logic and electronics concepts in a game-like environment", *Proc.of the Spring Simulation Multiconference.*, San Diego, CA. 2009.

[74] H.L. Jordan, "Saving Princess Sera Game Design, Usability, and Interface", Department of Computer Science, Lynchburg College, Lynchburg, VA. 2006.

[75] T. Barnes et al., "Game2Learn: A study of games as tools for learning introductory programming concepts", *Proc. of the 38th ACM Technical Symposium on Computer Science Education*, Covington, KY. 2007.

[76] E. Rossiou and S. Papadakis, "Educational games in higher education : a case study in teaching", *Proc. of the Int. Conf. on Education in a Changing Environment*, Salford, UK. 2007.

[77] E.O. Navarro and van der Hoek, A. "Comprehensive evaluation of an educational software engineering simulation environment", *Proc. of the 20th Conf. on Software Engineering Education e Training*, Dublin, Ireleand. 2007.

[78] S.B. Linek, D. Schwarz, M. Bopp and D. Albert, "When playing meets learning : methodological framework for designing educational games", *Web Information Systems and Technologies*, 45, 2010, pp. 73–85.

[79] M. Popescu et al., "Serious games in education", Proc. of 5th *European Conf. on Games Based Learning*, Athens, Greece. 2011.

[80] G. Sindre, L. Natvig and M. Jahre, "Experimental validation of the learning effect for a pedagogical game on computer fundamentals", *IEEE Transactions on Education*, 52(1), 2009, pp. 10–18.

[81] L. Natvig and S. Line, "Age of Computers − game-based teaching of computer fundamentals", *Proc. of the 9th Annual Conf. on Innovation and Technology in Computer Science Education*, Leeds, UK. 2004.

[82] L. Natvig, S. Line and A. Djupdal, "Age of computers", *34th Annual Conf. on Frontiers in Education*, Savannah, GA. 2004.

[83] R. Mittermeir et al., "AMEISE – A media education initiative for Software Engineering", *Proc. of the Int. Workshop on Interactive Computer Aided Learning*, Villach, Austria. 2003.

[84] J.M.D. Hill, C. K. Ray, J.R.S. Blair and C.A. Carver, "Puzzles and Games : addressing different learning styles in teaching operating systems concepts", *Proc. of the 34th Symposium on Computer Science Education*, Reno, NV. 2003.

[85] Binary Game, "The Cisco Learning Network", 2008. Available: https://learningnetwork.cisco.com/docs/DOC-1803.

[86] W. Wong and Y. Chou, "An interactive bomberman game-based teaching /learning tool for introductory C programming", *Technologies for E-Learning and Digital Entertainment*, Springer, Berlin, 2007, pp. 433–444.

[87] A. Hicks, "Towards social gaming methods for improving game-based computer science education", *Proc.of the 5th Int.Conf. on the Foundations of Digital Games*, Monterey, CA. 2010.

[88] CeeBot, 2008. Available: http://www.ceebot.com/ceebot/index-e.php.

[89] J. Singh, S.K. Dorairaj and P. Woods, "Learning computer programming using a board game – case study on C-Jump", Proc. of the Int. Symposium on Information and Communications Technologies, Kuala Lumpur, Malaysia. 2007.

[90] Colobot, 2001. Available: http://www.ceebot.com/colobot/index-e.php.

[91] M. Arevalillo-Herráez, et al., "Conquer the net: an educational computer game to learn the basic configuration of networking components", *Computer Applications in Engineering Education*, 20(1), 2009, pp. 72–77.

[92] T. Denning, T. Kohno and A. Shostack, "Control-Alt-Hack: a card game for computer security", *Technical report UW-CSE-12-07-01*, 1–7, University of Washington, *Seattle, WA.* 2012.

[93] V. Kern, M. do R. Stotz and M. Bento, "A game-playing experience in the learning of database concurrency control", Federal University of Santa Catarina, Florianópolis, Brazil. 2008.

[94] S. Ludi, "Software Engineering Unplugged", *Department of Software Engineering Rochester Institute of Technology*, Rochester, NY. 2010.

[95] R.O. Chaves et al., "Intelligent Behavior Simulation Module for Software Process Elements", *Proc. of the Brazilian Symposium on Games and Digital Entertainment.*, Salvador, Brazil. 2011.

[96] G.M. Soares and B.Z. Rausis, "Development of an educational game for teaching project management in undergraduate computing programs", *Project Thesis*, Information Systems, UFSC, Brazil. 2011.

[97] V. Srinivasan, K. Butler-Purry and S. Pedersen, "Using video games to enhance learning in digital systems", *Proc. of the Conf. on Future Play*, Toronto, Canada. 2008.

[98] L. Hakulinen, "DSAsketch: Data Structures and Algorithms drawing game", *Proc. of the 12th Koli Calling International Conf.on Computing Education Research*, Tahko, Finland. 2012.

[99] A. Chaffin, K. Doran, D. Hicks and T. Barnes, "Experimental evaluation of teaching recursion in a video game", *Proc. of the Symposium on Video Games*, New Orleans, LA. 2009.

[100] P. Léger et al., "Business simulation training in information technology education: guidelines for new approaches in IT training", *Journal of Information Technology Education*, 10, 2011, pp. 39–53.

[101] J. Ryoo, Techatassanasoontorn, A., Lee, D. and Lothian, J. "Game-based InfoSec Education using OpenSim", *Proc. of the 15th Colloquium for Information Systems Security Education.* Fairborn, OH. 2011.

[102] INNOV8, 2008. Available: http://www-01.ibm.com/software/solutions/soa/innov8/index.html.

[103] IT Manager, 2007. Available: http://itmanagerduels.intel.com.

[104] Kid Krypto. "Computer Science Unplugged", 2007. Available: http://csunplugged.org/public-key-encryption.

[105] Light-Bot, 2010. Available: http://armorgames.com/play/6061/light-bot-20.

[106] P. Kruchten and J. King, "Mission to Mars: an agile release planning game", *Proc. of the 24th Conf. on Software Engineering Education and Training*, Honolulu, HI. 2011.

[107] J. Srinivasan and K. Lundqvist, "A Constructivist approach to Teaching Software Processes", *Proc. of the 29th Int. Conf. on Software Engineering*, Minneapolis, MI. 2007.

[108] NET Terrarium. "NET Terrarium 2.0 project!", 2008. Available: http://terrarium2.codeplex.com/.

[109] M. Leverington, M. Yüksel and M. Robinson, "Using role play for an upper level cs course", *Journal of Computing Sciences in Colleges*, 24(4), 2009, pp. 259–266.

[110] Paper Tower Competition, 2009. Available: http://www.gqs.ufsc.br/paper-tower-competition/.

[111] J. Wein et al. "Virtualized games for teaching about distributed systems", *Proc. of the ACM Technical Symposium on Computer Science Education*, ACM New York, NY, 2009, pp. 246–250.

[112] J.M. Fernandes and S.M. Sousa, "PlayScrum - a card game to learn the scrum agile method", *Proc. of the 2nd Int. Conf.on Games and Virtual Worlds for Serious Applications*, Braga, Portugal. 2010.

[113] PM Master, 2012. Available: http://www.gqs.ufsc.br/pm-master/.

[114] A. Baker, E.O. Navarro and A.van der Hoek, "An experimental card game for teaching software engineering processes", *Journal of Systems and Software*, 75(1-2), 2005, pp. 3–16.

[115] Project Risk, 2001. Available: https://www.successfulprojects.com/Store/ProjectRiskBoardGame/tabid/68/Default.aspx.

[116] Project-o-poly (PoP), 2007. Available: http://www.semq.eu/leng/proimplog1.htm.

[117] C.M. Zapata and G. Awad-Aubad, "Requirements Game: teaching software project management", *CLEI Electronic Journal*, 10(1), 2007.

[118] T.C. Kohwalter,E.W.G. Clua and L.G.P. Murta, "SDM – An educational game for Software Engineering", *Proc. of the Brazilian Symposium on Games and Digital Entertainment*, Salvador, Brazil. 2011.

[119] J. Bell, S. Sheth and G. Kaiser, "Secret ninja testing with HALO software engineering", *Proc.of the 4th Int. Workshop on Social Software Engineering*, Szeged, Hungary. 2011.

[120] L.G.C. Hamey, "Using the security protocol game to teach computer network security", *Proc. of the Symposium on UniServe Science Improving Learning Outcomes*, Sydney, Australia. 2003.

[121] A. Drappa and J. Ludewig, "Simulation in software engineering training", *Proc. of the Int. Conf. on Software Engineering*, Limerick, Ireland. 2000.

[122] K. Shaw and J. Dermoudy, "Engendering an empathy for Software Engineering", *Proce. of the Australasian Conf. on Computing Education*, Darlinghurst, Australia. 2005.

[123] E.O. Navarro and A. van der Hoek, "SimSE: an interactive simulation game for software engineering education", *Proc. of the 7th Int. Conf. on Computers and Advanced Technology in Education*, Kauai, HI. 2004.

[124] R.S. Longstreet and K. Cooper, "Using games in Software Engineering education to increase student", *Proc. of the 24th Conf. on Software Engineering Education and Training*, Honolulu, HI. 2011.

[125] Simulate Computer Game, 2008. Available: https://sites.google.com/site/childrenandtechnology/presentation-3-simulate-computer.

[126] E. Magno et al., "SimulES: a game for teaching Software Engineering", *Project Thesis*, Computer Science, PUC, Rio de Janeiro, Brazil. 2006.

[127] A. Jain and B. Boehm, "SimVBSE : Developing a Game for Value-Based Software Engineering", *Proc. of the 19th Conf. on Software Engineering Education e Training*, Oahu, HI. 2006.

[128] O.H. Graven, H.A. Hansen and L. Mackinnon, "A computer game modelling routing in computer networks as abstract learning material in a blended learning environment", *Int.Journal of Emerging Technologies in Learning*, 2009.

[129] Subnet game. "The Cisco Learning Network", 2008. Available: https://learningnetwork.cisco.com/docs/DOC-1802.

[130] Subnet Troubleshooting Game, 2010. Available: https://learningnetwork.cisco.com/docs/DOC-7558.

[131] A. Dantas, M. Barros and C. Werner, "A simulation-based game for project management experiential learning", *Proc. of the 16th Int. Conf. on Sofware Engineering and Knowledge Engineering, Banff, Canada*. 2004.

[132] M.O. Barros et al., "Model-driven game development: experience and model enhancements in software project management education", *Software Process: Improvement and Practice*, 11(4), 2006, pp. 411–421.

[133] Waterfall Game, 2009. Available: http://www.the-software-tester.com/games.html.

[134] B. Tan and J.L.K. Seng, "Game-based learning for Data Structures : a case study", *2nd Int. Conf. on Computer Engineering and Technology*, Chengdu, China. 2010.

[135] C.W. Ford and S. Minsker, "TREEZ - an educational data structures game", *Journal of Computing Sciences in Colleges*, 18(6), 2003, pp. 180–185.

[136] G. Taran, "Using games in Software Engineering education to teach Risk Management", *Proc. of the 20th Conf. on Software Engineering Education e Training*, Dublin, Irleand, 2007, pp. 211–220.

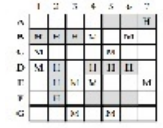
[137] G. Jiménez-Díaz et al., "Pass the Ball : Game-based Learning of Software Design", *Entertainment Computing*, *Lecture Notes in Computer Science*, 4740, 2007, pp. 49–54.

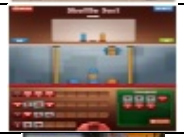
[138] Wireless Explorer. "The Cisco Learning Network", 2010. Available: https://learningnetwork.cisco.com/docs/DOC-7560.

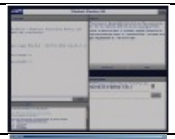
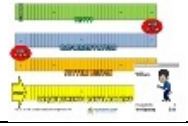
[139] C.G. von Wangenheim, M. Thiry and D. Kochanski, "Empirical evaluation of an educational game on software measurement", *Empirical Software Engineering*, 14(4), 2009, pp. 418–452.

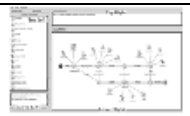
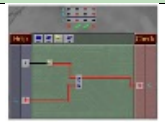
[140] A. Syberfeldt and S. Syberfeldt, "A serious game for understanding artificial intelligence in production optimization", *Proc. of the Conf. on Computational Intelligence and Games*, Copenhagen, Denmark. 2010.

TABLE II. GAMES FOR TEACHING COMPUTING

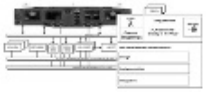
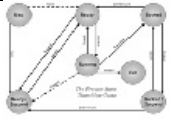
No.	Name	Computing Knowledge Area	Screenshot/Photo	Gameplay	Reference(s)
1	3DAR Lego Game	Human Computer Interaction		Players have to re-construct a 3D model using its decomposed pieces. The game provides 3 game levels with increasing difficulty. At level 1, all pieces are precise components of the model, at level 2 each piece is a primitive geometry (cube, cone, etc.) that has to be customized and at level 3 no pieces are provided and the players have to create the model from primitive geometries.	[72]
2	Age of Computers	Computer Science		Player travel through computer history from early mechanical computers, transistor age to embedded systems of the present age. They have to solve different kind of problems (multiple-choice, numbering, ALU control signals, etc.) to earn points. The availability of content in the game depends on the player's progress. A chat window for each historical period is used for communication with other students and teaching assistants.	[80] [81] [82]
3	Algorithms Recursive Game	Algorithms and Complexity		During the game, the players have to answer questions on different levels of difficulty covering various aspects of recursive algorithms. Players advance their tokens by rolling a dice. If a player arrives at the bottom of a ladder or at the head of a snake, a question is displayed. If the player answers correctly, her/his token moves up to the top of the ladder or stays at the head of the snake.	[76]
4	AMEISE	Software Engineering		AMEISE is based on SESAM. In the game, students assume the role of a technical project manager. They can hire and fire personnel, structure the project, and allocate tasks. Students are challenged to manage a project according to a particular model of the problem structure, selected by the instructor. It is up to the instructor to select the number of trials (simulation runs) to solve given tasks within specified constraints. Students can learn from previous simulation runs, change their strategies and measure their own success using the AMEISE self-assessment feature.	[83]
5	Anti-Phishing Phil	Security		The main character of the game is Phil, a young fish living in the Interweb Bay. Phil wants to eat worms so he can grow up to be a big fish, but has to be careful of phishers that try to trick him with fake worms (representing phishing attacks). Each worm is associated with a URL, and Phil's job is to eat all the real worms (which have URLs of legitimate web sites) and reject all the bait (which have phishing URLs) before running out of time. The game is split into four rounds, each one more difficult than the previous and focuses on a different type of deceptive URLs. Players have to correctly recognize at least six out of eight URLs within two minutes to move on to the next round. If a player loses all three lives, the game is over.	[66]
6	Battleships	Algorithms and Complexity		Using sheets with sequences of battleships, each player circles one of the ships. Then, the players have to guess where his/her partner's ship is. There are several variations of the game for different search algorithms (binary, hashing, etc.)	[37]
7	BattleThreads	Operating Systems		The class is broken down into one controller and two teams. The players are responsible for the placement of one ship each and for firing a shot from that ship each turn until their own ship is destroyed (or their side wins by sinking all of their opponents' ships). The controller gets the enemy team's ship layout and announces the effect (hit or miss) of each shot. At the end of a turn, the controller compares battle damage and reports the results. No other means of communications are allowed.	[84]
8	Binary Game	Net Centric		The game requires players to either set bits in a byte correctly to equal a given number or to insert the number represented by a given set of bits.	[85]

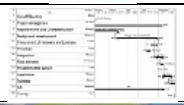
9	BINX	Digital Logic		The game takes place in the context of a computer being attacked by a virus. The main character is Chip, an integrated circuit designed with the purpose to wipe all traces of the malicious virus from the computer. Inside the computer, information is flowing from the motherboard to all output devices attached to the computer except the monitor. The adventure takes place inside the CPU. Players must navigate a path through the bus to find the viral infection plaguing the graphics processor and resolving several missions on different levels.	[41]
10	Bomberman Game	Programming fundamentals		In this game, players have to write C code to control the movement of the Bomberman.	[86]
11	BOTS	Programming Fundamentals		Bots is a game that contains a series of programming challenges that students must solve.	[87]
12	CARGO-BOT	Programming fundamentals		Players must program a crane to properly situate boxes using loops and conditional variables. It rewards players for discovering the most efficient way.	[54]
13	CEEBOT	Programming fundamentals		In CeeBot-A (for teenager and adults) the player has to program robots in outside environments on various planets. This version, deals, mainly with repetitions, conditional options, variables, arrays, functions and classes.	[88]
14	C-Jump	Programming Fundamentals		The goal of the game is to find the most efficient way to “ski” down a mountain. Therefore, players must make decisions based on common programming syntax to go down a certain path. By moving around the board, entering loops, branching under conditional and switch statements, the players gain physical experience of a complete program. First player to move all skiers past the finish line is the winner.	[89]
15	COLOBOT	Programming fundamentals		The player heads a space expedition and is assisted only by some robots. Her/his mission consists in successive attempts at the exploration and colonization of various planets. During the game, the player has to build and program new types of robots in order to complete the mission.	[90]
16	Computer Architecture Mini-Game	Computer Architecture and Organization		The player has to solve several problems on different levels of difficulty on logic gates and elements of computer architecture. The game starts with an initial score (1000 points) and each time a player makes a wrong decision the score is decreased. Textual hints are provided.	[67]
17	Computer Theory Jeopardy Game	Theory of Programming Languages		Similar to the Jeopardy game, players respond questions to win the game. Here, the questions are related to context free grammar and pushdown automata.	[84]

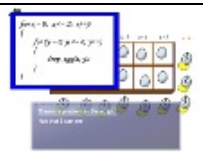
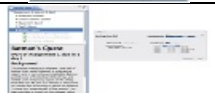
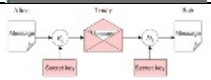
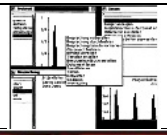
18	Computing Networks Mini-Game	Net Centric		The game encourages players to solve three different situations (one per level): how to connect different elements to have ADSL in home, how the information travels through the network, and how information is sent from a personal computer to another. Routers, IPs, Computer ports, browsers or frames are examples of puzzle pieces in the game. Correct actions lead to an increase of the player's score, which is reduced when making mistakes.	[67]
19	Conquer the Net	Net Centric		This risk-game based game takes place at a scenario composed of a number of PCs, switches, and routers, placed on a map. At the start of the game, a PC and an objective are assigned to each player. The game performs in turns. At each turn, a player is allowed to modify the configuration of any of the devices displayed on the board (e.g., change its IP address, mask, and gateway). The number of modifications a player may perform is determined by rolling a dice. The winner is the player who achieves his/her objective first.	[91]
20	Control-Alt-Hack	Net Centric		The players act as white hat hackers in a security consulting company. Each player is given a Hacker card. Gameplay is centered on missions—a variety of audit jobs and pro bono work that require the selective application of hacker skills: Hardware Hacking, Network Ninja, Cryptanalysis, Forensics, etc. The character's skill levels and player's dice rolls determine whether the player succeeds or fails at a mission. Players can increase their skill levels by purchasing useful items; opponents can hinder player's efforts to complete a mission by playing Lightning Strikes on them. Mission successes and failures lead to the gain and loss of Hacker Cred. Players win the game by accruing enough Hacker Cred and becoming the CEO of their own consulting company.	[92]
21	CounterMeasures	Security		The player is guided through several missions, each teaching a new aspect of security. Each mission has a title, a description, a score for completing the mission, a skill as the focus of the mission, objectives required to complete the mission, help given to guide the user, and a list of commands learned during the mission. The missions build upon each other, allowing the player to utilize previously learned skills in each new mission. Players are given a fully functional shell that runs commands while working on a mission.	[36]
22	CyberCIEGE	Net Centric		Players of this video game purchase and configure workstations, servers, operating systems, applications and network devices. They make trade-offs as they struggle to maintain a balance between budget, productivity, and security. In its longer scenarios, users advance through a series of stages and must protect increasingly valuable corporate assets against escalating attacks.	[33]
23	Database concurrency control card game	Information Management		In this game, students play cards and build a schedule for a given set of simultaneous transactions. They use their knowledge about transactions and concurrency control protocols to simulate the work of a transaction processing system. The sequence of activities is simulated through a control card or kanban that is passed from student to student. The card exchange is similar to a procedure call. Each student has to decide what to do at her/his turn, based on the concepts s/he learned and the concurrency control protocol being used.	[93]
24	Databases: Lots of Data and Getting Quick Results	Information Management	N/I	Players execute database operations by representing different roles, such as, user, timekeeper, DBMS, or data. The Data Record players stand in line and, then, simulating a query the DBMS goes to each record to check, if they are in the query. If yes, they move to the designated results area. The timekeeper states the time to conduct the search.	[94]
25	Dealing with difficult people	Software Engineering		During this game, groups of learners realize a project kick-off meeting. One of the group members represents the project manager, who moderates the meeting and has to make sure that in the end all members confirm their commitment. The problem is that each of the other group members is assigned a role of a difficult person being instructed to "act", e.g., as a whiner, no-person etc. Once the project manager reacts in an adequate way, the person turns into a cooperative member and gives her commitment to the project.	[48]
26	Deliver!	Software Engineering		Once the game starts, player pairs advance in the project execution rolling a dice - in addition risk events may happen. Each completed round all player pairs have to pay the weekly salary for their project teams. When finishing a project phase, the player pairs have to monitor the project progress calculating Earned Value indicators and may change the project plan (e.g., by firing or contracting new team members). Winner of the game it the player pair, who first delivers the product to the customer arriving at the field product delivery without running out of financial resources during execution.	[31]

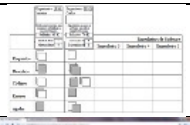
27	DesigMPS	Software Engineering		The player assumes the role of a process engineer and has to model a process based on given process descriptions (aligned to the Brazilian software process improvement model MPS.BR). The game offers 4 levels with increasing degree of difficulty in terms of given elements. The created process models are compared to pre-defined solutions and, based on their degree of similarity, a score is assigned. Goal of the game is to achieve maximum score.	[71] [95]
28	Detective Game – what killed the project?	Software Engineering		The game takes place in the context of a fictitious company that completed a software project for the development of an online pizza web site. The project failed and now the company is contracting the players as consultants in order to identify what went wrong. Therefore, the players receive a set of project documents. The players have to analyze the project documentation and track project progress by revising the weekly status reports applying Earned Value Management. For each correct calculation and correctly identified time or cost overrun, the players receive a point. The winner is the group of players that obtained the largest number of points.	[96]
29	Digi Island	Digital Logic		The game takes place on an exotic island, where 1s and Os in digital circuits are represented through usable and unusable spaces on the island. The player is an adventurer on the Digi Island to be transformed into a tourist attraction by developing real estates, such as, amusement parks. The goal of the player is to construct a minimum number of buildings as large as possible covering all the usable spaces, while satisfying some further regulations.	[40]
30	Digital Logic and Electronics Concepts	Digital Logic		The game is designed in levels progressing from basic to advanced topics on digital logic. The levels are separated into three groups. The first group focuses primarily on binary information, the second group focuses on logic functions, and the third group focuses on circuits with memory.	[73]
31	Digital System Game	Digital Logic		The game starts with the player in one corner of an imaginary 3D world similar to those found in first person-shooter games. The player's goal is to reach the exit, which can be accomplished by unlocking several doors and obtaining two skill upgrades. At each locked door, the player is presented with a sum-of-products combinational circuit problem. The game switches to a 2D environment for the digital circuit design problems. The game updates the external outputs automatically to indicate the values of the outputs of the current circuit for the specified input values.	[97]
32	DSAsketch	Algorithms and Complexity		Players are divided into two teams. The main idea of the game is to draw concepts related to data structures and algorithms on the whiteboard while other team members are trying to guess the concept in the picture. The winner of the game is the team with the highest score once all players had a turn to draw.	[98]
33	EleMental: The Recurrence	Programming Fundamentals		EleMental is a death match pitting players against one another in a winner-takes-all battle. Players can manipulate terrain, wind, water, and temperatures as they fight for supremacy against their opponents applying programming. 3D FPS style game that runs on both PC and Xbox 360.	[99]
34	ERPsim	Information Systems		Using a continuous-time simulation, students are put in a situation in which they have to run their business with a real-life ERP system. Students, thus operate a company; be it a bottled water distributor, a make-to-stock cereal manufacturer, or a dairy company, and must interact with suppliers and customers by sending and receiving orders, delivering their products and completing the whole cash-to-cash cycle.	[100]
35	Groupthink /Second Life	Software Engineering		Multi-player Second Life version of the groupthink exercise game focusing on requirements engineering. The objective of the game is to test the ability of a group of learners to reach consensus on software specifications. After discussing the specification within the group, players individually answer a set of questions on the specifications and the system evaluates the number of corresponding answers and presents performance statistics. Winner of the game is the group with the highest score at the end.	[47]
36	hACME Game	Security		The game is organized as a series of levels where the player must overcome a set of challenges in order to unlock access to the next level. Each level focuses on a set of well-known security vulnerabilities.	[62]

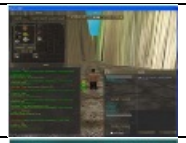
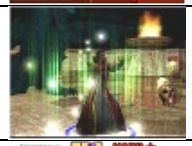
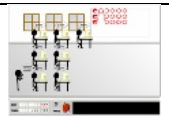
37	Immersive Security Education Environment (I-SEE)	Security		The learning module simulates a scenario in which two teams are competing against each other: one acting as a group of attackers and the other acting as a group of defenders. Each team has a base consisting of a packet assembly board, a building containing a money counter and a console, and a pipeline representing a network connection from the base to the gateway router. The game begins, when each team successfully creates a data packet according to the TCP/IP protocol on the packet assembly board. As soon as frames start moving, the other team can launch an attack, while the team has to defend the attacks.	[101]
38	INNOV8	Information Systems		The player interacts with other virtual employees, participating in their daily activities in a fictitious company. During the game, the player will be involved in three types of Business Process Modeling activities: process discovery and process modeling, collaboration-driven simulation and iterative process improvement and real-time business management.	[102]
39	IT Manager	Information Systems		Using a collection of technologies (cooling, security, mobility, multi-core machines) the player has to keep the servers running and the people working. The problems users face range from virus attacks or slow computers to broken disks or strange beeps. The problems the player has to resolve range from simultaneous support requests to spam in their inbox.	[103]
40	JV2M	Programming Fundamentals		The game takes the form of a 3D videogame in which the player must compete to provide the right machine instructions, collect resources needed by the instructions and use her/his knowledge about Java compilation to find the best strategy to win the game. This means that a player has to compile and execute byte code quicker than the opponent.	[63]
41	Kid Krypto	Security	N/I	Players send secret messages using encryption to pass information on a treasure map.	[104]
42	Lego Factory	Intelligent Systems (AI)		The simulation shows a physical production process in the form of a miniature factory for producing "bubble gums" using Lego bricks and Mindstorms NXT. As the factory production manager, the player's task is to find the best configuration (wrt. production flow and buffer sizes) possible for the production process in order to maximize profit. The player wins the game, if the optimal configuration of the factory is found. In order to find optimal solutions, players have to develop AI algorithms.	[140]
43	Lego Tower Team Activity: Managing Change	Software Engineering	N/I	Players are instructed to construct Lego towers. Winner of the game is the group that constructs the largest tower. During the construction, changes are requested (e.g., only one person can construct - other team members can only give instructions, tower has to be put on wheels to be mobile).	[94]
44	Light-Bot	Programming Fundamentals		In this game, the player has to program a robot to light up blue tiles. Each time the objective is achieved, the player advances to the next level.	[105]
45	MIS Project Manager (Formerly known as the Information Systems Project Manager Game)	Software Engineering		The player takes the role of project manager in an IS development project. Once selected a strategy for focus, the player works through the systems development life-cycle. In addition, s/he has to deal with events occurring during the project execution.	[27]
46	Mission to mars - Release or iteration planning	Software Engineering		The game is a Monopoly-style board game, game illustrating the planning process in iterative software development. Some factors such as uncertainty in estimation, actual velocity, and occurrence of defects are simulated by rolling a dice. Hard constraints and dependencies between stories are added to stimulate discussion on the strategy to pursue and how to mitigate risks.	[106]
47	myPlanNet Simulation	Net Centric		During the game, the player represents the CEO of a service provider, who must manage the company as it evolves. The player has to connect citizens with the next-generation IP network and guide them into the Connected Life with the wonders of visual networking.	[34]

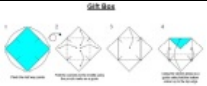
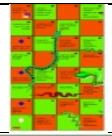
48	Next Generation_7 helicopter game	Software Engineering		The game is played for a minimum of three iterations by project teams that are each composed of sub-teams covering the functional areas of requirements, design, implementation, integration, and reserve funds. Each member of the project team receives the problem statement, the cockpit avionics architecture diagram and a detailed set of instructions about their own role. The first iteration of the game, for example, is set up as a classical waterfall model, wherein the different teams work in a sequential fashion with discrete handoffs between the phases.	[107]
49	.NET Terrarium 2.0	Programming fundamentals		Applying the .NET programming model and languages, the player can create herbivores, carnivores, or plants and then introduce them into a peer-to-peer, networked ecosystem where they compete for survival. Once the creature is loaded into Terrarium, it acts on the instructions supplied by its code.	[108]
50	Operating systems role plays	Operating Systems	N/I	In the game, students assume the roles of processes, while the instructor represents the processor. The game itself involves running sample sets of program code in a step-wise fashion so that students can see the actions and consequences of each segment of code.	[109]
51	Paper Tower Competition	Software Engineering		During the game, students have to build a paper tower for a given set of requirements while executing a systematic project management process, including the application of Earned Value Management for monitoring & controlling the project. Winner of the game is the group that manages to build the tower and obtained the largest SPI and CPI with minor total project cost taking into consideration also the beauty of the design of the tower.	[110]
52	Path	Programming Fundamentals		Paths string together a series of questions that students must complete in sequence so as to complete a path. They may either be multiple choice questions or more general coding questions where automated public and private test cases are run to verify the students' answers.	[53]
53	PDConsole	Distributed Systems		The basis of the game is a barebones, but fully functional integrated website that combines features of social networking and video sharing. The site displays (mock) advertisements when different pages are viewed. The game generates artificial traffic against the site, and the overall system metric of health is the number of advertisements served and resulting revenue. The game is a sort of fire fighting exercise that begins when the game administrator breaks parts of the system in some way. Players notice that the system's performance has degraded because ads and revenue drop. Then, they have to identify the problem and fix it.	[111]
54	PlayScrum	Software Engineering		PlayScrum is derived from Problems and Developers. It is a card game in which each student plays the role of a Scrum Master in a software development project adopting SCRUM practices. The game is divided into sprints that differ from project to project and during which each player must develop a number of tasks. The winner is the player, who first performs all tasks without errors or who completed the highest percentage of tasks without errors at the end of the last iteration.	[112]
55	PM Master	Software Engineering		Trivia-style board game with multiple-choice questions about software project management on different knowledge areas, such as, scope, time and quality management in alignment with PMBOK (4.Ed.). The player, who first responds correctly one question of each of the nine project management knowledge areas, wins the game.	[113]
56	Problems and Programmers	Software Engineering		A card game that simulates the software process from requirements specification to product delivery based on the waterfall life cycle. Players take the role of the project leader in the same project and compete to be the first to complete the project. They pass through the phases of the software process and draw cards and take actions to continue the development as well as to react to problems. The winner of the game is the player who first achieves a sufficient number of integrated code cards without bugs.	[24] [114]
57	Process State Transition	Operating Systems		Each group of players is given a game board representing the seven-state process transition model. One of the students is selected to be the operating system (OS), one the timekeeper (TK), and the others become programs, each keeping track of some number of processes as they are managed by the operating system. When a process moves into a state in which it must be present in memory (Ready, Running, etc.), the player who owns that process places its memory markers on a grid representing available memory. When the process is suspended, the memory markers are lifted, indicating that the process has been moved out of main memory.	[84]

58	Programming Fundamentals Mini-Game	Programming fundamentals		The player has to solve several problems on different levels of difficulty. The game provides the player with a skeleton of the code where s/he then has to place the different code pieces. The game start with an initial score (1000 points) and each time a player makes a wrong decision the score is decreased. Visual color-coded hints are provided.	[67]
59	Project Execution Game (PEG)	Software Engineering		In this game, the players work in groups in order to manage a project. They receive a detailed project plan and the success evaluation criterion, which is to complete the project within the minimum possible budget. Project overruns due to players' decisions, cause penalties and overhead cost, which negatively affect the team's success.	[50]
60	Project Risk	Software Engineering		The board game is based on a path through the project management lifecycle. A project manager pawn progresses through the path over 12-rounds of project play (representing project reporting periods). Progressing steps may cost or return chips. Six game pawns represent team members (that can be lost through certain project risks). When playing competitively, the winner is the one who finishes with most chips and team members left.	[115]
61	Project-o-poly (PoP)	Software Engineering		The goal of the game is to achieve the highest possible profit by renting, buying and selling the projects located around the game board, until becoming the richest Project Manager (PM) and, possibly, the monopolist. Starting from "Go!", every PM moves his/her token on the game board by rolling the dice.	[116]
62	Requirements Activity: Lego House	Software Engineering	N/I	In the beginning, a project is described by the instructor (client) for building a house. The players have 30 minutes to provide a house that meets the client's needs. Throughout the game, the players can elicit requirements three times with the client asking three questions each time. At the end, the groups present the constructed houses in order to check, if they satisfy the client's requirements.	[94]
63	Requirements collection and analysis game (RCAG)	Software Engineering		The basic idea of the game is for a team to manage and deliver a number of software development projects. Each player has a specific role such as project manager, systems analyst, systems designer or team leader. Several project scenarios are available with underlying business models that define budget, schedule and resources. The player(s) assigned to the system analyst role has to identify the requirements for the project.	[26] [20]
64	Requirements Game	Software Engineering		The players assume different roles (director, analyst, designers, and programmers). The instructor coordinates the game execution and acts as customer. During the game, the groups have to develop a requirements specification, including, e.g., a model of the proposed solution, entity-relationship diagram and relational model. Raw materials, computer time and team member work time are restricted. The game can be played in 1- or 2-cycles depending on the available time.	[117]
65	Robot Trouble	Programming Fundamentals		The player is supposed to navigate a robot through a maze of traps to reach the destination by programming its movements.	[59]
66	Robozzle	Programming Fundamentals		RoboZZle is a programming puzzle game. The player has to program a robot to collect all items from 2D tri-colored field.	[60]

67	Saving Princess Sera	Programming Fundamentals		Saving Sera is a 2D exploratory game, where the player has to rescue the kidnapped princess Sera. Therefore, the player has to create “machine” (which are programs) to solve problems. In Saving Sera, the player fixes the machine by unscrambling a while loop to track a fisherman’s catch; debugging a nested for loop placing eggs in crates; and visually piecing together a flowchart for quicksort. When the player makes a mistake, s/he must fight a script bug by answering various computer science questions.	[32]
68	Scrum Lego City	Software Engineering		Players execute sprints building LEGO houses and vehicles from user stories following the SCRUM process and performing SCRUM ceremonies.	[55]
69	SDM - Software Development Manager	Software Engineering		In this game, the player has a team of employees to develop software. The game presents possibilities to the player to decide on development strategies and to define roles for each staff member. When the software is completed and delivered to the customer, there is a quality assessment of the software and a project completion payment.	[118]
70	Secret Ninja Testing	Software Engineering		HALO presents a series of quests of software engineering tasks. Quests can either be individual, requiring a developer to work alone, or in groups, requiring a developer to form a team and work collaboratively towards their objective. Completing each quest gives the students experience points and achievements. Through a global leader board students can track their own and other students' achievements and experience points.	[119]
71	Secure Volunteer Game	Net Centric		In the game, the player joins as "volunteer" an organization with the task to enable connectivity. The player sets up VPN connectivity to enable relief workers in the field to securely share environmental, health and medical information with the headquarters, while moving through virtual rooms in the game and interacts with fictitious characters to configure a VPN concentrator.	[35]
72	Security Protocol Game	Security		Within each group, one player is selected to play Alice or Bob, the two communicating parties. Another player is selected to play Gavin. The same player may also take the role of Colin. The remaining player(s) take the role of Trudy the intruder. The game starts with the players seated around a table. The students select a game scenario to play, e.g., Alice wishes to purchase computer software from Bob over the Internet using her credit card for payment, and a protocol to use such as the Transport Layer Security protocol.	[120]
73	SESAM	Software Engineering		Students get a project of a given size assigned. They take the role of a project manager, aiming to complete the project within given time and budget by a team of simulated, virtual software engineers. Students can hire personnel form a pool of persons with different qualifications and different expected salary. The player controls the simulator using a purely textual interface.	[121]
74	SharkWord	Software Engineering		The game creates a virtual environment in which projects develop in (accelerated) real time. The player is forced to act to problems immediately and intervene properly. The game is propelled by an underlying suspense story in the context of building a shark aquarium. The game covers not only economic aspects, but also social aspects, conflict management and diplomatic skills.	[30]
75	SimjavaSP	Software Engineering		The student, acting as project manager, has to develop a simulated software project within the required time and budget, and of acceptable quality.	[122]

76	SimSE	Software Engineering		A customizable simulation environment for educating students in software processes/management. The environment supports the creation and simulation of game-based software process simulation models (e.g., waterfall, incremental, XP). In the game, learners take on the role of the project manager and must manage a team of developers in order to successfully complete a software project or task in conformance to the respective software process model.	[123] [29]
77	Simsoft Game	Software Engineering		Simsoft players are formed into teams of 2 or more and they are given a scenario that describes the requirements for a small software development project. Taking the role of project manager, the team must manage the project from start-up to final delivery. The players gather around a printed game board to discuss the current state of the project and to decide on their next move. The board shows the flow of the game while plastic counters are used to represent the staff of the project.	[25]
78	SimSYS	Software Engineering		The game world represents a software development organization (offices, meeting rooms). The player starts play as Scrum Master for an agile development process. S/he is presented with a product description that needs to be developed for a product owner. The player must select a diverse, adequately skilled scrum team, elicit requirements and create a release backlog with the product owner and, then, manage development sprints (short iterations).	[124]
79	Simulate Computer Game	Computer Architecture and Organization	N/I	Every player assumes a role (processor, mouse, etc.) identified by props (big nerd glasses, helmet etc). Then, simple functions are executed by players.	[125]
80	SimulES	Software Engineering		SimulES is based on the Problems and Programmers game introducing a game board to organize the cards. Similar to Problems and Programmers, the players´ objective is to be the first to complete a pre-defined software project. A player performs different roles such as software engineer, technical coordinator, quality controller and project manager.	[28] [126]
81	SimulES-W	Software Engineering		SimulES-W is a digital version of the game SimulES. The game allows a student to take on different roles (project manager, auditor, software engineer) in a project to build software, allowing him/her to experience common tasks and decisions in the context of software development.	[28]
82	SimVBSE	Software Engineering		The game starts with a visit to the CEO's briefing room, where the student through animated videos is asked to take on the role of a project manager, and briefed on the current organizational situation and the student's overall objective in the game. Making a move in the game involves visiting different rooms (board room, lounge, etc.) and choosing one or more of the available options.	[127]
83	Sorting Algorithms	Algorithms and Complexity		In this game, the student has to collect boxes and sort them according their values by using sorting algorithms (Bubble and Bucket Sorting).	[39]
84	SortingCasino	Algorithms and Complexity		SortingCasino resembles the card game Casino. On a turn, player can capture one or more cards from the table by using one of his or her hand cards. With an algorithm card, player can capture all special cards that are valid for that algorithm. Respectively, with a special card, player can capture all algorithm cards that are valid for the criteria in the special card and put them on his/her victory stack. The game ends when either algorithm stack or special card stack is empty. The winner is the player with most cards in her/his victory stack.	[38]

85	SortingGame	Algorithms and Complexity		SortingGame is a card game that includes two decks of cards: algorithm cards and special cards. Each algorithm card has a name of one sorting algorithm on it. Special cards contain criteria that either apply or do not apply for a given algorithm. In the beginning, the dealer deals 3 algorithm cards and 2 special cards to each player. The actual game round consists of two phases: special card phase and algorithm phase. During the special card phase, each player can place one special card to the table. In algorithm phase, each player places one algorithm card to the table. The algorithm should be valid for the active special cards on the table. The winner of the round is the player whose algorithm's asymptotical time complexity is the best. The winner collects all the cards played during the round and adds them to his/her victory stack.	[38]
86	Starter MMO	Net Centric		The game combines routing and forwarding. Students start out in an area, marked start on a map representing a maze. In this area there is a Non Player Character (NPC) that gives them quests to perform. These quests all consist of delivering a package (representing a 'data packet') to another NPC (host) somewhere. The students are neither told where the destination is located nor about the layout of the maze. The aim is to run to the NPCs location (simulating transmission of the data packet), and deliver the package to them.	[128]
87	Subnet Game	Net Centric		This game allows the player to solve a number of IP subnetting problems in top secret Area 51. Each level must be solved in an allotted amount of time.	[129]
88	Subnet Troubleshooting Game	Net Centric		The job of the player is to identify problems in otherwise correct subnets and fix them so the network works properly. When all problems in a level have been solved, the player moves to the next level. The objective of the game is to complete the problems in all the levels in the game and get the highest score possible.	[130]
89	The Catacombs	Programming Fundamentals		In this 3D fantasy game, the user is an apprentice wizard who must cast three increasingly complex spells (programs) to save two children who are trapped in the catacombs. The first spell uses IF statements to magically unlock a door, the second uses nested FOR loops to construct a bridge, and the third spell uses nested FOR loops to solve a cryptogram.	[75]
90	The Hard Choices	Software Engineering		The Hard Choices game board represents activities of a software development release. In their quest to become the market leader, players are competing against each other to release their product to the market place. Players earn points for landing on a square with a tool (representing rewards for investing in technical infrastructure) or by not finishing last (representing rewards for speed to market). When a player crosses a hard choices square, s/he must decide whether to go over the shortcut bridge or to go the long way and try to collect one or more tool cards.	[49]
91	The Incredible Manager	Software Engineering		In the game, the learner acts as a project manager being responsible for planning, executing, and controlling a software project. The goal is to complete a project, whose cost and schedule are established during a planning phase and approved by stakeholders. Project execution occurs in continuous turns, consuming the planned resources. The learner must monitor the project execution and take corrective actions, when necessary. Visual effects and reports provide feedback, showing exhausted developers, late tasks, etc.	[131] [132]
92	The MIS Game	Software Engineering		Key elements are a board with tokens representing progress; artifacts the player can “own” (e.g., money); “events” that happen and the dice, which “drives” progress. Players are given a certain budget at the start and are required to acquire and deploy resources to develop a portfolio of systems. Then the player focus on the IS development.	[27]
93	The Mystery of Traffic Lights	Digital Logic		Designed from a first-person perspective, the game starts at the major intersection of a small town, where an engineer character, Jack, is standing frustrated by the busy and messy traffic due to malfunctioning traffic lights. He then invites students to help him redesign the controller with the right logic for the current traffic flow.	[42]

94	The Orange game	Net Centric		Players are labeled with a letter of the alphabet and for each player two oranges are marked with the player's letter. The oranges are distributed equally to the players (except one who receives only one orange). Players pass the oranges around until each child gets the oranges labeled with their letter of the alphabet.	[56]
95	The Software Development Game	Software Engineering		Players must build origami boxes with one of the following four groups of letters, SO, FT, WA or RE. Every box represents a software module. One group of four modules forms one software piece (a complete word, SOFTWARE). Every module must accomplish a set of pre-defined requirements, which can be discussed with the director of the game. The goal, therefore, is that the players must compete in groups to gain profits from an imaginary software company that makes software modules.	[58]
96	The Waterfall Game	Software Engineering		The game is a snake and ladders-style game in which players take a turn to throw the dice and advance on the game board.	[133]
97	Tower of Cubes	Algorithms and Complexity		A list of cubes in the tower represents the data in a stack or queue depending on the selected mode (Stack or Queue). When the game starts, 5 cubes with two random colors are dropped into the tower. Subsequently, a new cube with a random color is dropped into the tower automatically either to the top/rear of the tower depending on the mode. When two consecutive cubes with the same color meet, both cubes will be disappeared. The player needs to move the cube in and out in order to clear the cubes in the tower and the score will be increased. When all cubes are cleared in the tower, the player wins.	[134]
98	Treasure Hunt	Programming fundamentals		The goal of the game is to get the pirate o the treasure by defining his treasure map. Therefore, the player has to drag commands from the panel on the right to the main command area and then execute the program to have the pirate follow the created "treasure map".	[61]
99	TREEZ	Algorithms and Complexity		In the single-player version, a player is given a randomly generated binary tree with labeled nodes. Then, s/he must traverse the tree as if he/she were an ant crawling on the tree's surface with the object to move through the tree in the correct order. Time restrictions are imposed on individual moves, and the player gets to compete against his/her previous fastest time. In the two-player version, each player is given a different tree. Then, players alternate turns.	[135]
100	Software Risk Management Game	Software Engineering		In the game, each player assumes the role of a project manager. The game's objective is to develop a product, sell it in the market and win by having more money at the end of the game than all the other players. The game has 5 stages: planning, requirements, architecture & design, implementation and testing. Yet, unlike traditional board games, players are free to roam around, choose where to go and what to do, without the need for throwing a die to advance.	[136]
101	U-Test	Software Engineering		The game is based on a case in which the player is considered a candidate for a tester job in a software company. After a job interview, the player must solve challenges by preparing unit test cases. The game presents the player's score based on his/her performance.	[68]
102	ViRPlay3D2	Programming Fundamentals		Students are immersed using a first-person view that simulates the point of view of the objects. An aiming point in the center of the screen serves to interact with other entities in the virtual world and to represent message passing. The user interface also displays a score that contains information about the current state of the game session In the single-player mode, the user does not perform the role of any object, but s/he is an external avatar that observes and partially controls the game. In multi-user mode, ViRPlay3D2 represents a virtual space where users discuss the interactions among objects needed to carry out a scenario execution.	[137]

103	Welcome to SCRUMIA	Software Engineering		The game deals with the planning and execution of a sprint applying SCRUM where the students have to produce paper boats, hats and planes. Each group member takes on a specific role (SCRUM master, etc.). During the game, the students execute the SCRUM process, including sprint planning, daily meetings, etc. Winner of the game is the group that obtained the highest profit and business value.	[57]
104	Wireless Explorer	Net Centric		Players compete by ensuring friendly space aliens remain in wireless communication throughout their galactic voyages. Welcoming an envoy of alien scientists, players have to configure open wireless access to a space ship mainframe correctly for each alien. Individual and team scores are based on successful site survey strategies, deployment skills and network maintenance.	[138]
105	Wu's Castle	Programming Fundamentals		Players interact in two ways: by manipulating arrays changing loop parameters and by physically walking the game character through loop execution. After an introduction to the game story and interface, the player walks through a visual representation of for-, while-, and do-while-loops. The player then manipulates a one-dimensional array by setting the parameters in a for loop. The player repeats this process with a nested loop walkthrough and two-dimensional array manipulation.	[32]
106	X-MED	Software Engineering		In this game, the learner takes the role of a measurement consultant in a software organization. During the game session, the player passes sequentially through all steps of a GQM-based measurement program creating measurement artifacts based on a series of constrained selections with pre-defined alternatives. For each decision, the game gives a feedback and score. At the end a total score and feedback report is provided.	[139]
107	Z-Buffer	Computer Architecture and Organization		In the game a series of image fields is given on which a student should click to determine the value of each bit, i.e. contents of the registry in various situations presented in the task. The correct answer to fill the content of one bit is one of the proposed answers presented to students in the form of squares to be selected.	[65]