AN EXPERIMENT WITH AN OPTIMIZATION GAME

Ela Pustulka¹, Thomas Hanne¹, Benjamin Adriaensen¹, Stefan Eggenschwiler¹, Egemen Kaba¹
and Richard Wetzel²

¹*FHNW, School of Business, Institute for Information Systems, Olten, Switzerland*

²*Lucerne School of Information Technology, Rotkreuz, Switzerland*

ABSTRACT

We aim to improve the teaching of the principles of optimization, including computational intelligence (CI), to a mixed audience of business and computer science students. Our students do not always have sufficient programming or mathematics experience and may be put off by the expected difficulty of the course. In this context we are testing the potential of games in teaching. We deployed a game prototype (design probe) and found out that the prototype led to increased student motivation, intuitive understanding of the principles of optimization, and strong interaction in a team. Ultimately, with the future work we sketch out, this novel approach could improve the learning and understanding of optimization algorithms and CI in general, contributing to the future of Explainable AI (XAI).

KEYWORDS

Computational Intelligence, Optimization Game, TLBO Algorithm

1. INTRODUCTION

We teach optimization in a master level business information systems course alongside business process management, IT management and the cloud, digital economy, business intelligence, and security (FHNW 2019). The optimization course focuses on computational intelligence (CI) approaches and other metaheuristics. Student feedback tells us that some students opt out, as they think they do not have sufficient skills. We want to address such student perceptions by using games to lower the entry barrier. Our scientific interest is the desire to understand the complexity of the learning process and to gain knowledge on how algorithm teaching is best conducted. We presented our course design and the teaching methods in (Dornberger & Hanne 2011, Dornberger & Hanne 2017, Hanne & Dornberger 2017). A broader context for this work is provided by the fields of explainable AI (XAI) (Gunning 2016) and educational game research (Johnson et al. 2016, McGill et al. 2017a, McGill et al. 2017b, Petri & von Wangenheim 2018). On the one hand, computational intelligence games which teach advanced algorithmic techniques contribute to XAI, as they let people experience algorithms in action and further the understanding of computerized decision making. On the other hand, we know educational games have been used as an effective tool for maths teaching (Garneli 2017), but we do not know if this can be transferred to optimization and how to do that as the usefulness of games in optimization teaching has not been studied so far. We want to discover how games could be used in teaching future business leaders, decision makers and engineers who are not all expert programmers or mathematicians. This has the potential to give all students and the general public the ability to learn, use and interpret AI methods, particularly optimization, via challenging games, or, possibly, a mixture of digital and analogue games, visualization, and instruction.

We conducted a successful experiment in teaching the principles of population-based optimization via a game. We observed that an optimization game conveys the idea of optimization successfully while keeping the students motivated and focused on solving the task as a team. The rest of the paper is structured as follows. Section 2 discusses related work. Section 3 presents the methodology and Section 4 the results. In Section 5 we discuss this work, outline our future research and conclude.

2. RELATED WORK

Our work spans optimization, explainable AI and educational games. The Explainable AI (XAI) program (Gunning 2016) has two goals: to use the existing knowledge of the psychology of explanation to deliver a computational model of explanation supported by software; and to design explainable learners allowing us to interpret decisions made by AI tools. This will make sure algorithmic decisions can be properly understood, explained and trusted. ACM, on the other hand, has issued a Statement on Algorithmic Transparency and Accountability (Association of Computing Machinery 2017) which includes seven principles: awareness, access and redress, accountability, explanation, data provenance, auditability, and validation and testing. XAI is a response to new laws, such as the EU General Data Protection Regulation 2016/679 (Calder 2018).

Against this background, we are interested in games for education and knowledge (Schrier 2016), variants of which are called serious games (Abt 1970). This research field can be split into two main camps. The proponents (Gee 2007, Hamari et al. 2016, McGonigal 2011, Schrier 2016) believe that games revolutionize teaching and improve learning (van Gog 2013, Wouters et al. 2013). The critics, however, say that there is little evidence that games improve the educational outcome (Clark et al. 2010). Mayer (2016) promulgates the use of games closely related to the educational content and including personalization, a conversational style and speech instead of text, providing feedback and coaching. He calls for adaptive games which develop as the student learns, to keep motivation high. Kriz (2007) characterizes research into gaming simulation as representing two distinct fields: the design sciences and the analytical sciences. In analytical sciences a game can help to test, justify, or develop theories and can produce new universal knowledge, as in protein structure prediction (Cooper et al. 2010). In design sciences the focus is on design and usability studies which tell us how to design a game that appeals to users and how far the game serves the needs of the users, educators, policy makers or a business in the process of change. In mathematics teaching, there are strong arguments in favor of games (Kiili et al., 2014, McLaren et al., 2017). Also, business schools use simulations to teach business concepts, such as enterprise resource planning (ERP). Legner et al. (2013) report on adapting their course to use an existing simulation and observe clear cognitive gains from simulation and a significant learning effect, with a good understanding of ERP, KPIs and the balanced scorecard.

Foldit, for instance, is a serious optimization game which falls outside the educational agenda. It supports scientists in protein structure modeling (Cooper et al. 2010, Cooper 2014). Players generate new protein models which are then integrated in protein prediction pipelines. The game is based on the following assumptions: a large number of players who like solving 3D puzzles, problems that need a solution requiring human ingenuity (3D matching and modeling skills), a scoring function which has to be minimized (energy function), and scientists who can use the solutions generated by the players to improve their protein structure models. Scoring is analogous to solving complex optimization problems: there is no best score and only ranking is possible. There is strong social interaction and the players are mostly motivated by purpose, immersion and achievement (Cooper 2014). Foldit revealed that players have various strategy preferences and develop many novel valid approaches to solve problems. Players develop recipes and the game supports those via a block-based visual programming interface or scripting. Foldit inspires and informs our work.

3. METHODOLOGY

We follow the design research method (Barab & Squire 2004, Vaishavi et al. 2004). The method situates research in context, considers the complexity of real life situations, uses flexible designs, focuses on social interactions and multiple aspects of design in practice and is interdisciplinary. As part of the awareness phase, we conducted an interview with a part-time Master student attending the optimization module. The goals were: to capture the student's experience in learning algorithms in a traditional lecture environment, to find out how he studies, to learn about his experience with games and about his attitude towards using a game in the future. The student confirmed that when learning algorithms, it is important that one deals with them in depth. The reading of papers and other sources helps to get a first understanding, but an algorithm can only be fully understood by implementing it. The student said that the lectures were highly theoretical and offered little interaction between the student and the lecturer, which is typical of university teaching in many subjects. This made reviewing at home very important. As additional sources, the student used online forums or tech-blog entries. He has tried to read the suggested papers, but found them to be hard to understand. He was comfortable

with lecture-based teaching. Due to his technical background (BSc Computer Science), he had no significant problems in understanding algorithms. However, he had not played educational games so far. He stated, he could imagine that a well-developed game that conveys the learning content in an interesting way could certainly be successful.

In teaching optimization, we follow a number of steps representing a growing mastery of the subject.

STEP 1. Realizing the need for algorithmic optimization.

STEP 2. Optimizing manually.

STEP 3. Using a standard optimization solution.

STEP 4. Understanding a number of algorithms and the problems they fit.

STEP 5. Choosing a suitable algorithm, encoding the problem, solving it, and reflecting.

STEP 6. Extending an algorithm, tuning it, and comparing the outcome to the standard version.

All steps can benefit from the use of games. For instance, Step 1 may use an existing "Slick Oil Distribution Game" (Hoorman 2017) which clearly shows that the problem is not easy to solve using pen and paper. Such a game is static and does not demonstrate the use of an algorithm. Standard optimization solutions (step 3) are available from logistics and supply chain management vendors. The issue with most such software is that we have no knowledge which algorithm is being used and cannot modify it. Using open-source software solutions (steps 4 to 6) is often the preferred way but has three prerequisites: understanding the principles of optimization, being able to program algorithms, and understanding some mathematics.

The Teaching-Learning-Based Optimization (TLBO) (Rao et al. 2011) was selected, as it naturally led to a game story involving teaching and learning. TLBO is culture inspired and performs well in engineering and science contexts (Rao 2016). TLBO has a population which takes part alternately in teaching and learning. The design variables are represented as subjects the students take and the student grades in all subjects represent their fitness. The teacher is selected as the best learner so far. Learning happens in pairs. The TLBO consists of the following steps.

INITIALIZATION: optimization problem (function), population, design variables (subjects and grades) with constraints, termination criterion

TEACHER PHASE:

- a. calculate the population mean of each design variable
- b. identify the teacher (best score)
- c. modify each student based on teacher and mean, using a random teaching factor tr_i
- d. if the solution is better, accept it.

STUDENT PHASE:

- a. randomly divide students into pairs
- b. in each pair: let the weaker student learn from the better student, using a random learning factor lr_j
- c. if the solution is better, accept it.

TERMINATION test (criterion): if satisfied, output the best solution, else start TEACHER PHASE.

Recent work on educational game evaluation comes from Petri et al. (2018), who, however, do not focus on game design. Their model has nine dimensions: usability, confidence, challenge, satisfaction, social interaction, fun, focused attention, relevance, and perceived learning. This mirrors the Octalysis framework (Chou 2015), which we selected, as it supports game development. Octalysis lists eight core motivational drives and specifies how they can be broken down into specific game elements. The drives are: meaning, accomplishment, empowerment, ownership, social influence, scarcity, unpredictability and avoidance. The model sees player experience in terms of two types of outcome: the white hat core drivers enhance well-being and the black hat core drivers provide immersion and can lead to addiction. The model distinguishes between extrinsic and intrinsic drivers, assuming that intrinsic drivers are more effective. The approach links to the classic storytelling concept of the Hero's Journey (Chou 2015). Since improved learning can be linked to intrinsic motivation, our game development focuses on the core drives that directly relate to mastery, autonomy, purpose, and relatedness. Mastery (accomplishment) requires that one gets feedback on game progress. Autonomy (empowerment) is the player's ability to decide on the next action to be taken. Purpose (epic meaning and calling) in the game is contributing to common good. Finally, relatedness (social influence) is based on cooperation and team work. The playing process has four phases. Discovery allows the players to learn why they are playing (the story and purpose). During onboarding the players learn the rules of the game. During scaffolding they build up playing skills and get feedback about the decisions they make. During endgame the goal is achieved. The game should motivate during all four stages and various motivation mechanisms may be needed at various phases.

We chose the TLBO algorithm and designed the game, by applying the Octalysis framework focusing on intrinsic motivation. Then a game with a software back and front end was developed and tested (Adriaensen et al. 2018). Our experimental design comes from design based research (Barab & Squire 2004), as we wanted to assess the feasibility of our approach and to identify the design gaps. The game was evaluated in a two part experiment, with 4 experimental subjects (randomly assigned to two groups). The subjects were not told beforehand what the experiment was about. Their educational backgrounds did not include computer science or optimization. In the first stage (45 min.) two subjects played the game, while at the same time two other subjects studied the optimization algorithm from a two page description and discussed it. In the second stage (30 min.) a non-structured group discussion with four participants was held. Both parts were captured on video.

4. RESULTS

PAPER STUDY. Two male subjects (avg. age 32.5), an MSc student and a BSc student, participated. The master student had a background working with algorithms, and the bachelor student did not. They received a 2-page description of the algorithm and asked to study and discuss it. One of the first reactions of this group was the inquiry about the winning condition. Although there was no gaming/winning incentive, an interesting dynamic in the group developed in which an active shift of knowledge was attempted.

The timeline for this group consists of three phases: *reading* and understanding; *formulation* of the algorithm; and the *exchange of knowledge*. During *reading* we observed silence, concentration and a few frowns and notes on paper. Both participants showed the desire to learn. The paper was the focus of attention. Signs of autonomy and mastery were observable. The *formulation* phase was different. The more knowledgeable participant talked and walked around. The whiteboard was discovered and used to formulate and draw ideas and became the focus. Learning was supported by drawing, talking, and reading, which demonstrated autonomy. However, the autonomy of the BSc student decreased with time, as the MSc student took over control. His engagement and motivation were high. Relatedness rose as the participants started interacting. During *knowledge exchange* (teaching) the moderator tried to confuse the situation in order to raise the engagement of the bachelor student. The MSc student took over the whole process at this point, which led to a teacher/student dynamic. This could also be classified as a sign of purpose as the master student felt the need to explain the algorithm. The MSc student had fully mastered the topic. However, the BSc student struggled with it. At this point, his interest and focus were at its lowest point.

We interpret this as follows. The MSc student had experience of studying scientific papers. He processed new information efficiently, and quickly built a mental model of the algorithm. The BSc student did not have enough background, either technical (algorithms) or scientific (reading papers), to effectively process information. The experiment showed uneven motivation and engagement and serves as a baseline for the second experiment using a game.

GAME EXPERIMENT. The game was designed and implemented as part of an innovation project in an MSc in Business Information Systems by three students (coauthors) using design thinking techniques (Lewrick et al. 2018). The focus was on designing a game that would get the players to understand the principles of population-based optimization. The story involves a population stranded on a desert island, with a plane and no crew, as the crew perished. The way to escape is to repair the plane and learn how to fly it back to safety while controlling fuel consumption. The players assign the passengers to the Campfire (learning), the Plane (teaching) or to Upskilling (stargazing, tinkering, or building sandcastles). Upskilling increases fitness (a small preset reward), whilst gradually nudging the player into discovering that the teaching-learning mechanics are much more effective in improving the score. The learning goal is to realize that population-based optimization can quickly generate a solution with a good score. The game mechanics were designed so that the player can only win if she uses the teaching and learning activities in turn.

The game was implemented in JavaScript. We modified the TLBO algorithm to convey just the essence of population-based optimization. The population size (passengers) and the generation number (days) are set to 10. There are four design variables (engineering, piloting, mechanics, navigation). To motivate the players and keep them guessing the rules, we added Upskilling and Idling. We adapted the algorithm so that at the same time some characters are in the teaching part, some in the learning part, and the rest idling or upskilling. We sum up the four design variables (all with max=25, sum=100). We had two proto- typing sessions and adjusted the design, as simulation showed the need for improvement. We realized the players needed feedback about

the fuel efficiency (total score). We numbered the figurines, for easy identification and added an application showing each figurine's skills (Figure 2), to replace scorekeeping on paper. This allowed for fluid play and improved the game experience.

Figure 1. Left: the game with the figurines, the island, and the activities. Right: the game being played

Figure 2. Skill levels (0-25) for one character: green – improving, red – worsening, gray – no change

The game starts when the players sit down at the table with the scenery: an island with a palm tree, five illustrated activity options, and ten small figurines, see Figure 1. The moderator introduces the game by telling the story and the rules, guidelines on how to interact with the characters, and the game objective. From the explanation and the handout, the players learn that they will manage a group of stranded passengers. To escape, the survivors have to learn to use the plane to fly off the island and reach the civilization. To do so, the plane's fuel efficiency needs to be managed. This requires skills that the group currently does not have to the required degree, such as engineering, mechanics, piloting and navigation, which all positively contribute to fuel efficiency. Each turn is a new day where the players allocate the survivors to one of three hubs: Campfire, Plane or Upskilling and select the upskilling activity. The game front end, Figure 2, tells the players how the characters are learning. The goal is to reach a fuel efficiency of 80% within 10 turns. As the players place the figurines on the island activities, the moderator updates the console. She also tells the players what the current fuel efficiency is (the value of the function to be optimized).

Two female subjects (avg. age 26), both holding Master degrees, formed a team. During the introduction both showed interest in the game and asked questions showing motivation (mastery drive), search for meaning, and autonomy. Even before the game started, one suggested that there will be a "pattern to discover". We observed live social interaction. The players discussed strategy, efficiency and possible game mechanics, which showed relatedness and social influence. They formed a strategy: to inventory the skills and assess the impact of various activities. We saw this as empowerment and accomplishment/mastery drive. In rounds 1-2 the participants were playing happily. They were enthusiastic and amazed when they saw their skills improve. After the first round they recognized that the skills evolved differently when the characters were assigned to the campfire (learning). However, they kept their strategy of trying out all activities. By the end of round 2, the excitement peaked. The participants asked for a piece of paper to note their observations, revealing immersion. We saw that not providing writing material was a design weakness. In round 3, the players showed slight disappointment. As they experienced frustration, the social layer became visible, as they interacted more to generate new approaches. One player said that she expected that the characters would have to be moved around constantly, unknowingly referring to the optimization algorithm dynamics. Starting in round 4, the allocation of characters to activities seemed random. The players asked about the upskilling behavior. In round 5, one

participant felt "lost". In round 6, the players were told that there was a learning and a teaching activity. This did not help, and the players still aimed at understanding the game. In round 9, the players were told what the teaching and learning activities were. One recognized the influence of the characters on each other's skills at the campfire. When the last round was finished, and the players did not win, they were disappointed, which points to the need for mastery/accomplishment. When asked whether they would like to play again, they replied very positively. The end game was reached in round 11, and a player spontaneously exclaimed that it was an optimization algorithm. The session finished with a brief explanation of the algorithm.

We had to modify the game during the experiment. First, the players were given a pen and paper they asked for (round 2). Soon after the game start we realized we need to impose a time limit for a turn. Then we had a software issue which interrupted the game for 15 minutes, with the subjects asked not to discuss the game during the break. We saw that onboarding was not needed as the game presented no usability problems and the participants were absorbed in the game right from the start.

In terms of the game phases, we saw *exploration* where the players were trying to understand the game mechanics and *discovery*, as they were examining the skill evolution in the game. However, during the 10 rounds we did not see *focusing* which would involve testing the game to maximize the results, as the players did not manage to build a mental game model. In the 10th round, however, they moved to solving with the help of the moderator. We saw that discovery and onboarding happened organically, while scaffolding did not.

During the following non-structured group discussion with all four subjects, the players indicated that they were very engaged and motivated to win the game and that the learning content was valuable. When presented with the description of the algorithm, Player A did not understand one of the visualized explanations used in the paper the study group used, even though the mechanics of optimization were obvious to her after playing the game. This is a very strong indicator that the game significantly lowered the barrier towards integrating the content. The game made the concept of optimization understandable. Another participant added that one learns by failing and recovering, and this dynamic was part of the game. The discussion also revealed some flaws. Player B mentioned that a more extensive explanation of optimization algorithms would be required in order to improve the learning experience. Additionally, the calculation of fuel efficiency was not transparent. This might have led to the occurrence of the "clueless phase" in play rounds 6 to 10. Additionally, the group mentioned that the calculation of the upskilling needed tuning and that the game was too complex.

SUMMARY. We studied two different optimization learning paradigms: self-study and game. The self-study mirrors our current understanding of how optimization can be taught effectively (in the classroom) or less effectively (via self-study or group study). It showed that the less experienced student had too little experience in self-study and in algorithm understanding, and therefore could not master the algorithm, even though the more experienced student was trying to help him. Clearly, an experienced teacher transmits the required knowledge effectively.

The game investigated a new paradigm of optimization teaching. It demonstrated that a game can be an excellent introduction to the topic and can convey an intuitive understanding of the TLBO, without the need to study the code and learn the theory first. The game scenario had a high appeal and brought in strong learning motivation. The first two rounds of the game were very effective. We saw intrinsic motivation to learn, engagement and enjoyment. We expected that to be followed by a scaffolding phase where the players improve in playing and progress in their understanding. However, it turned out that our prototype gave the players too little feedback and they could not construct a mental model of the game. The game was too complex and had too many distractions. We now think, just providing two activities (teaching and learning) in a simpler game would have been more effective. The help given by the moderator in round 6 did not help the players. Once the players were told which activity was learning and which teaching, they started generating a model, but as the game was limited to 10 turns, the players failed to reach the goal of 80% fuel efficiency. So, the number of turns we foresaw was not sufficient to allow for good learning, and the players won by being allowed to do one more turn, in round 11. The prototype needs a moderator, which could be automated in a digital game, to provide for flow. Also, an improved visualization of scores and overall fuel efficiency is needed. In a teaching scenario, we believe, a player should not have time constraints, so that they can totally focus on the game.

The experiment was done collaboratively, in teams of two. This was a good decision, as strong team dynamics were observable, with emotional and intellectual engagement, and the desire to win as a team. This forced the players to talk aloud, express their feelings and observations and share the experience. However, we realize that a game which supports self-study and reinforcement of classroom material, which we envisage as another option, also needs to be supported, and a future game has to allow both team or single-user play, with exchange of strategies and thoughts via a group room, like in Foldit (Cooper et al. 2010).

We note that the effort to construct the game and carry out the experiment was considerable. There is little guidance in the literature on how to design educational optimization games, as this is a new topic, and there are no toolsets we could reuse to build a game quickly, which points to the need for a tooling and methods infrastructure.

5. DISCUSSION AND CONCLUSION

Our starting point was the difficulty of attracting business students to study CI, optimization and algorithms. A teaching solution we propose is to use games to make optimization, and AI in general, more accessible. Our experiment demonstrated that a game can convey optimization ideas while making learning fun in a social context. Next research phases will improve the games, add more algorithms and play scenarios and test them in real life contexts, including both teaching and business and policy making.

To achieve the goals, supporting methods and toolsets are needed. We envisage a conceptual architecture consisting of: a software framework, a set of optimization problems and matching games, a theoretical framework supporting educational goals and teaching strategies, motivation, and evaluation, practical tools including a teaching handbook, ideation cards for game design (Wetzel et al. 2017), and a toolset for game development.

A software framework will support game construction from primitive blocks. This will shorten the development as it will be reused. To derive a framework, we can use an optimization API (Dornberger et al. 2010). We will build an abstraction over the gameplay layer, and a learning level abstraction allowing us to create games with various motivational mechanisms, teaching aims, and strategies. The research will be based on an underlying theoretical framework covering the educational, motivational, and evaluation perspectives.

In summary, we designed and tested an optimization game based on the TLBO algorithm. This first game motivated the players and led to some intuitive understanding of the algorithm. The analysis shows that further research in this direction is needed, and we outline the components for future work, consisting of a software framework, a theoretical framework, games, and tools and materials supporting ideation, game construction, and teaching. We believe optimization games have the potential to reach the student population that might otherwise miss out on the skills our future requires.

REFERENCES

- Abt, C.C., 1970. *Serious games: The art and science of games that simulate life*. Viking Press, USA.
- Adriaensen, B., Eggenschwiler S., Kaba, E., 2018. Gamification of an Optimisation Algorithm, ITP Project Report, FHNW, <https://github.com/EgemenKaba/TLBO-Game>, <https://github.com/StefanEggenschwiler/TLBO-GameClient>.
- Association of Computing Machinery, 2017. *Statement on Algorithmic Transparency and Accountability by ACM U.S. Public Policy Council, approved January 12, 2017 and ACM Europe Policy Committee, approved May 25, 2017*. https://www.acm.org/binaries/content/assets/public-policy/2017_joint_statement_algorithms.pdf.
- Barab, S. & Squire, K., 2004. Design-Based Research: Putting a Stake in the Ground. *Journal of the Learning Sciences* Vol. 13, No. 1, pp 1–14. <https://www.jstor.org/stable/1466930>.
- Calder, A., 2018. *EU GDPR: a pocket guide*. IT Governance Publishing. <https://www.jstor.org/stable/j.ctt1cd0mkw>.
- Chou, Yu-Kai, 2015. *Actionable Gamification: Beyond Points, Badges and Leaderboards*. CreateSpace.
- Clark, R.E. et al, 2010. An analysis of the failure of electronic media and discovery-based learning: Evidence for the performance benefits of guided training methods. In *Handbook of training and improving workplace performance*, Silber, K.H. et al. (ed.). Vol. 1. Pfeiffer, pp 263–297. <https://doi.org/10.1002/9780470587089.ch8>.
- Cooper, S., 2014. *A framework for scientific discovery through video games*. Morgan & Claypool Publishers.
- Cooper, S. et al, 2010. Predicting protein structures with a multiplayer online game. *Nature* Vol. 466, No. 7307, pp 756–760. <https://doi.org/10.1038/nature09304>.
- Dornberger, R., Hanne, T., Frey, L., 2010. The Way to an Open-Source Software for Automated Optimization and Learning - OpenOpal. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC*, pp 1–8. <https://doi.org/10.1109/CEC.2010.5586123>.

- Dornberger, R. & Hanne, T., 2011. E-Learning and Problem-Based Learning in Teaching Information Systems - Changing the Style of Teaching in the Information Systems Programs. In *Proc. of the 2nd International Conference on Education, Training and Informatics (ICETI)*. http://www.iiis.org/cds2011/cd2011mc/iceti_2011/paperspdf/eb183ga.pdf.
- Dornberger, R. & Hanne, T., 2017. Problem-Based Learning in Teaching the Module 'Optimization for Business Improvement'. In *ICSIT 2017 - 8th International Conference on Society and Information Technologies, Proceedings, 2017: 75–79*. <http://www.iiis.org/CDs2017/CD2017Spring/papers/EB808AV.pdf>
- FHNW 2019. University of Applied Sciences and Arts Northwestern Switzerland, <http://www.fhnw.ch/en/>, on 29.5.2019.
- Garneli, V. et al, 2017. Serious games as a malleable learning medium: The effects of narrative, gameplay, and making on students' performance and attitudes. *British Journal of Educational Technology* Vol. 48, No. 3, pp 842–859.
- Gee, J.P., 2007. *Good video games + good learning: Collected essays on video games, learning, and literacy*. Peter Lang.
- Gunning, D., 2016. *Explainable artificial intelligence (XAI)*, Defense Advanced Research Projects Agency, Broad Agency Announcement. <https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf>.
- Hamari, J. et al, 2016. Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior* Vol. 54, pp 170–179.
- Hanne, T. & Dornberger, R., 2017. *Computational Intelligence in Logistics and Supply Chain Management*. Springer.
- Hoormann (Lewis), S., 2017. Simple Game for Teaching the Value of Optimization, *Blog on Supply Chain Digest*. <http://bit.ly/2xLagDY> consulted on 28.03.2019.
- Johnson, C. et al, 2016. Game Development for Computer Science Education. In *Proceedings of the 2016 ITiCSE Working Group Reports (ITiCSE '16)*, Arequipa, Peru. ACM, pp 23–44. <https://doi.org/10.1145/3024906.3024908>.
- Kiili, K. et al, 2014. Studying the User Experience of a Tablet Based Math Game. *International Journal of Game-Based Learning (IJGBL)* Vol. 4, No. 1, pp 60–77. <https://doi.org/10.4018/IJGBL.2014010104>.
- Kriz, W.C., 2017. Types of Gaming Simulation Applications. *Simulation & Gaming* Vol. 48, No. 1, pp 3–7.
- Legner, C. et al, 2013. Designing Capstone Courses in Management Education: Knowledge Activation and Integration Using an ERP-based Simulation Game. In *34th International Conference on Information Systems ICIS*, Milano, Italy. Association for Information Systems. <https://aisel.aisnet.org/icis2013/proceedings/ISCurriculum/8/>.
- Lewrick, M. et al, 2018. *The design thinking playbook: mindful digital transformation of teams, products, services, businesses and ecosystems*. Wiley.
- Mayer, R.E., 2016. What Should Be the Role of Computer Games in Education? *Policy Insights from the Behavioral and Brain Sciences* Vol. 3, No. 1, pp 20–26. <https://doi.org/10.1177%2F2372732215621311>.
- McGill, M. et al, 2017a. If Memory Serves: Towards Designing and Evaluating a Game for Teaching Pointers to Undergraduate Students. In *22nd ACM Conference on Innovation and Technology in Computer Science Education*, Bologna, Italy, pp 25–46. <https://doi.org/10.1145/3174781.3174783>.
- McGill, M. et al, 2017b. Game Development for Computer Science Education. In *22nd ACM Conference on Innovation and Technology in Computer Science Education*, pp 385–385. <https://doi.org/10.1145/3024906.3024908>.
- McGonigal, J., 2011. *Reality is broken: Why games make us better and how they can change the world*. Penguin.
- McLaren, B.M. et al, 2017. A Computer-Based Game that Promotes Mathematics Learning More than a Conventional Approach. *International Journal of Game-Based Learning* Vol. 7, No. 1, pp 36–56.
- Petri, G. & von Wangenheim, C.G., 2017. How games for computing education are evaluated? A systematic literature review. *Computers & Education* Vol. 107, pp 68–90. <https://doi.org/10.1016/j.compedu.2017.01.004>.
- Petri, G. et al, 2018. MEEGA+, Systematic Model to Evaluate Educational Games. *Encyclopedia of Computer Graphics and Games*. Springer, pp 1–7. https://doi.org/10.1007/978-3-319-08234-9_214-1.
- Rao, R.V., 2016. Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. *Decision Science Letters* Vol. 5, No. 1, pp 1–30.
- Rao, R.V. et al, 2011. Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design* Vol. 43, No. 3, pp 303–315. <https://doi.org/10.1016/j.cad.2010.12.015>.
- Schrier, K., 2016. *Knowledge games: How playing games can solve problems, create insight, and make change*. JHU Press.
- Vaishnavi, V. et al., 2004. Updated 2015. *Design Science Research in Information Systems*. <http://desrist.org/design-research-in-information-systems/>. Online. Association for Information Systems.
- van Gog, T., 2013. Time on task. In *International guide to student achievement*, J. A. C. Hattie & E. M. Anderman (ed.). Routledge, pp 432–433.
- Wetzel, R., et al. 2016. Developing Ideation Cards for Mixed Reality Game Design. *DiGRA: Transactions of the Digital Games Research Association* Vol. 3, No. 2. <http://todigra.org/index.php/todigra/article/view/73>.
- Wouters, P. et al, 2013. A meta-analysis of the cognitive and motivational effects of serious games. *Journal of educational psychology* Vol. 105, No. 2, 249. <https://psycnet.apa.org/doi/10.1037/a0031311>.