M

Modeling of Games and Game Strategies

Tiziana Margaria[1,2] and Anna-Lena Lamprecht[3]
[1]Department of Computer Science and Information Systems, University of Limerick, Limerick, Ireland
[2]Lero – The Irish Software Research Centre, Limerick, Ireland
[3]Department of Information and Computing Sciences, Utrecht University, Utrecht, Netherlands

Synonyms

Computational thinking; Low-code application development; Model-driven development; No-code application development; Problem-based learning; Problem-based learning putational thinking; XMDD

Introduction

Developing, managing, evaluating, and comparing strategies for problem-solving are central skills when handling demanding and complex problems, both in computer science and many sectors of the economy. Such strategies often exist in the mind of single individuals and are often based on intuition, experience, and personal knowledge. Especially when working in teams, it is a big challenge to find common means to clearly formulate, communicate, and evaluate strategies. Games and simulations are widely used nowadays to train specific skills in teams. Notable examples span a growing variety of professional areas. In emergency medicine and disaster management, for instance, the online European Master in Disaster Management (https://www.dismedmaster.com) features the use of simulations and interactive media as an essential technology to bring together its global cohort of participants and have them cooperate in groups to solve complex collaborative scenarios. Crew training for aerospace missions, pilot training, and assessment have included for a long time simulations and games, and many new approaches to smart and connected health, smart manufacturing, and collaborative design use games or game-like simulations to train situated behaviors of increasingly complex nature. Such approaches have been called *serious games*, or applied games, for games designed for a primary purpose other than pure entertainment. This term is now widespread, although the traditional gaming industry dislikes it.

It has been confirmed by various studies that games and simulations have a positive impact on achieving learning goals (Vlachopoulos and Makri 2017). As remarked in (Juan et al. 2017), “the evolution of computer technologies, together with the increasing speed of Internet-based communications, has promoted the use of simulation software and serious games in higher education.

A. Tatnall (ed.), *Encyclopedia of Education and Information Technologies*,
https://doi.org/10.1007/978-3-319-60013-0_209-1

These technological and methodological tools can significantly enrich the learning experience in almost any knowledge area. In particular, they will have a significant impact on how the learning process is performed in the so called Science, Technology, Engineering, and Mathematics (STEM) education." Furthermore, games typically make attractive and engaging course topics for students of all ages (Juan et al. 2017); on balance, results indicate that games and/or simulations have a positive impact on learning goals (Vlachopoulos and Makri 2017) and game development allows students to experience various aspects of software development by systematically exploring the mechanics of a game. Modeling games and game strategies with **dedicated modeling environments** instead of implementing them with conventional programming languages or by just using pre-existing game engines focuses the attention on problem-solving. Students concentrate on understanding and managing the domain-specific essentials of the strategy without having to deal with the technicalities of a programming language and coding. Not much literature is available yet on model-driven or model-based game strategy design and game development, so it is still difficult to quantify and give a systematic picture of their use in educational practice.

This entry presents and discusses three selected examples of short courses described in recent literature that use modeling of games and game strategies as a teaching tool (section "Modeling of Games and Game Strategies as a Teaching Tool: Examples"). It then discusses aspects relevant to the educational context in which they are used (section "Discussion/Evaluation"). Section "Conclusions and Outlook" concludes the entry with a summary and perspectives for the future.

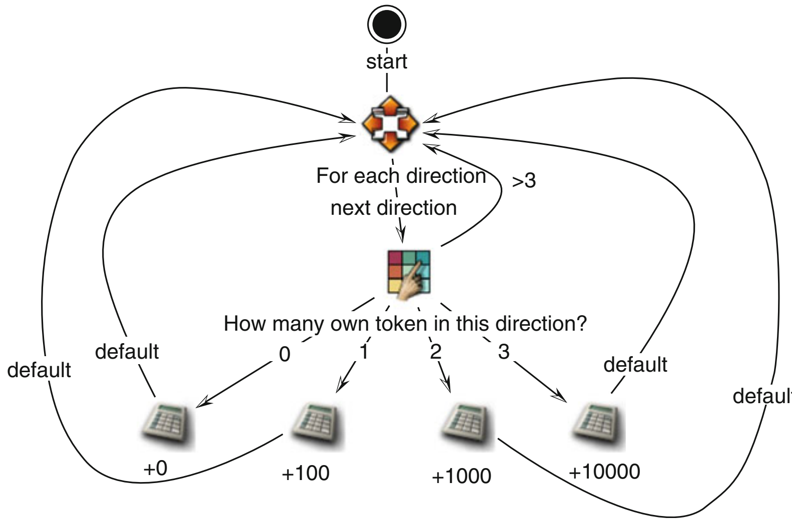
Modeling of Games and Game Strategies as a Teaching Tool: Examples

This section reports on three examples of game and game strategy development that have been successfully used as modeling tools to teach computational thinking concepts, for which detailed accounts can be found in recent literature. The three examples are the ConnectIT game, used as a computational thinking approach to strategy design (Bakera et al. 2009); the ChainReaction game, used successfully in summer camps for girls (IEEE 2016; McInerney et al. 2017); and a virtual world application to serious games in business and entrepreneurship based on Second Life (Science in 3D – BMBF-Projekt; Lehre in Virtuellen Welten).

They all use the XMDD paradigm (eXtreme Model-Driven Development) (Margaria and Steffen 2008) introduced first with the jABC modeling framework (Steffen et al. 2006; jABC) and later in DIME (Boßelmann et al. 2016). The paradigm is summarized in the entry "eXtreme Model-Driven Development (XMDD) technologies as a hands-on approach to software development without coding" of this Encyclopedia (Margaria and Steffen). jABC is a framework for model-driven development of service-oriented applications. It provides an intuitive yet powerful representation of processes in terms of service orchestrations (see Fig. 1) that use libraries of domain-specific building blocks as a reusable DSL. Applied to game strategies, this means that simple yet immediately functional and nontrivial example strategies can be created from a taxonomy of pre-defined domain-specific building blocks. In this entry they concern games, but other domains in use include scientific workflows in bioinformatics, business processes, text analyses in computer linguistics, healthcare processes, workflows in smart manufacturing, and several industrial applications in projects. After instantiation, each such component can be parameterized depending on the current context of use.

ConnectIT

ConnectIT (Bakera et al. 2009) was developed in 2008–2010 using jABC and repeatedly used in week-long CS summer camps with school pupils (grade 10–12), to teach computational thinking by means of strategies for the well-known connect-four game: Two players throw in turn a token in a board of seven columns and six rows. The token falls down to the lowest unoccupied position in



Modeling of Games and Game Strategies, Fig. 1 A simple strategy for ConnectIT ("Average Joe"). (From Bakera et al. 2009)

the column. Each player plays with its own token color. The player that gets a group of four connected tokens (either horizontally, vertically, or diagonally) wins the game. The game is drawn when the board is full and no player won. Strategies for this game were first studied by Allis (1988), who followed a knowledge-based approach. In ConnectIT, it is deliberately not possible to build the perfect strategy identified by Allis. Instead, the provided building blocks allow for the construction of strategies that analyze patterns on the board, and the goal for the students is to build the best (imperfect) strategy to win the competition with their peers.

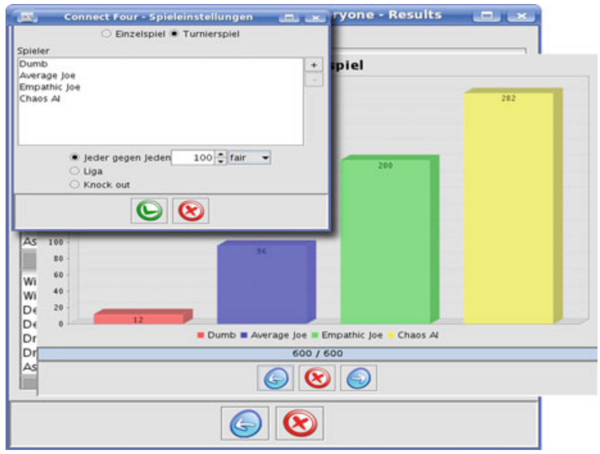
Figure 1 shows a simple and intuitive yet powerful representation of a strategy called *Average Joe*: It will give preference (expressed by a higher score) to cells that already have tokens of the same color nearby, thus increasing the chance of getting four in a row. Technically, the strategy is an orchestration of services that are the building blocks of a ConnectIT DSL. Once a strategy is created, having a measure at hand that allows for both concise and easy evaluation and comparison of several strategies offers two benefits:

1. First, a fast validation technique of the overall outcome reveals errors at an early development stage.
2. Second, the learning curve to become comfortable with the strategy under development is greatly improved – since strategies' behavior and impact depend on highly context-sensitive fine-tuning of parameters.

In the connect-four game, considering the adversary's possible moves additionally to the own ones also requires deciding which (relative) weight this information carries in the decision of the next move. Accordingly, ConnectIT provides an integrated evaluation functionality of the strategies and of their parametrizations in terms of tournaments. The effects can be studied in depth by playing several different kinds of tournaments (round robin, everyone against everyone, direct elimination, etc.) with (virtual) users that adopt the one or the other strategy. Each tournament returns a graphical presentation of the results (see Fig. 2 for a snapshot) and details about each match in the tournament. Combining several evolution steps in strategy development into a tournament enables students to directly witness and quickly pinpoint the effects of the decisions taken and to improve their strategy. This capability directly reveals whether a supposed enhancement of the strategy also yields a significant improvement of the strategy's competitiveness.

ChainReaction: Computing Camps for Girls

Increasing the number of female students in ICT-related university courses has been a major concern for several years. In 2015, McInerney



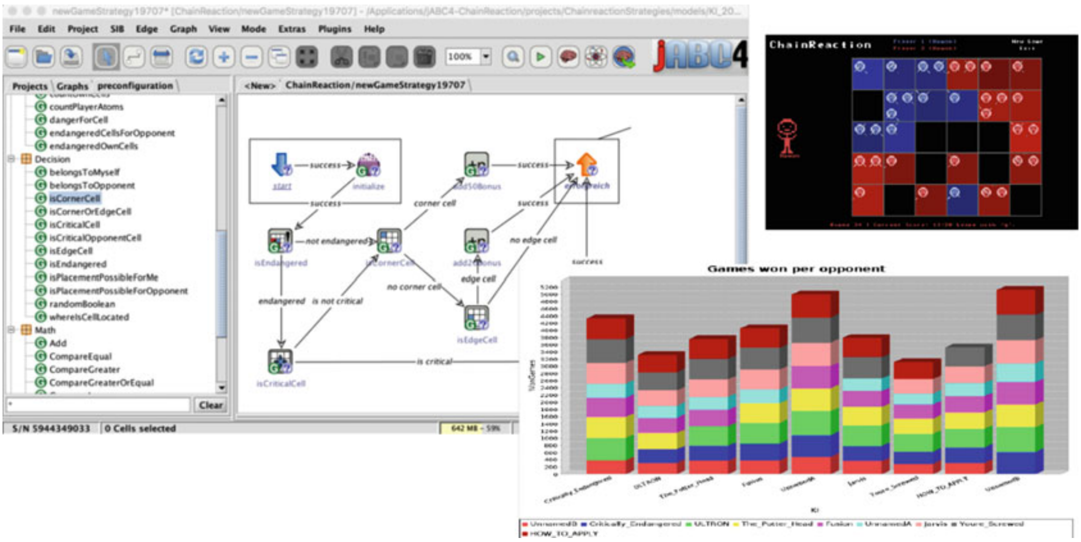
Modeling of Games and Game Strategies, Fig. 2 Evaluation of a ConnectIT strategy and snapshot of the tournament model. (From Bakera et al. 2009)

et al. offered for the first time a girls-only computing summer camp, as a new component in the education and outreach activities to foster students' interest in computer science (IEEE 2016; McInerney et al. 2017). Two 3-day camps for postprimary girls aged 14 and up were ran during the summer of 2015 at the University of Limerick. In the schedule for 3 days, the development of game strategies for the ChainReaction board game occupied ca. 2 days, including the students' final presentations. The other sessions were devoted to invited speakers from the industry and academia and to a robotics session with the NAO robot.

Modeling executable strategies for the ChainReaction board game was a major component of the program (blue sections in the table). ChainReaction (http://chainreaction.freewarepoint.de) is a strategy game for two players, who play against each other on a special 6x5 board, both trying to initiate chain reactions of atoms to conquer the whole board and win the game (see, e.g., the "Introduction to ChainReaction" page (https://hope.scce.info/chainreaction/) for a detailed description of the rules). The development of the game strategies to be followed by a computer player takes place in a specially prepared version of the jABC modeling framework (Steffen et al. 2006; jABC), which supports the intuitive, graphical development of flow-graph structures to define the behavior of a system. Figure 3 gives an impression of how the modeling of ChainReaction strategies with jABC works: The component library on the left contains the building blocks from which the strategies are assembled. The strategies focus on defining how the score for a cell on the board is to be computed. The framework then evaluates the strategies for all cells and places the next atom into the cell with the highest score (or, if there are multiple cells with the same highest score, randomly choosing one of them).

The framework, which is very similar to the successful jABC version for the modeling of game strategies for the more popular ConnectIT board game described earlier in this entry, combines an easily accessible application domain with a quick sense of achievement: The first game strategies are typically ready to run after a 30-minute introduction to the modeling tool. Students can then work on incrementally improving this starting strategy. They also get immediate feedback about the quality of the results by playing against their own strategies or by letting the strategy play against other computer players. As such, it provides a very motivational framework and a fun and lightweight learning-by-doing way to acquire and practice computational thinking skills.

In the first camp session, participants were introduced only to the basic ChainReaction game and had some time to play and familiarize themselves with it. The girls paired up on one



Modeling of Games and Game Strategies, Fig. 3 The ChainReaction game, modeling of ChainReaction strategies with jABC, and results of a tournament played between nine strategies

computer for this exercise and worked together in these pairs for the rest of the camp.

The next session was designed to foster more strategic thinking about the game. The task for the girls was to write down – simply on paper – how they had actually played the game, that is, which strategy they had followed to decide where to place their atoms. The groups then exchange strategies on paper, and each group tried to play strictly according to what the other group had written down. In a very entertaining way, this demonstrated the importance of formulating precise and unambiguous instructions.

The last session of the first day introduced the modeling tool and taught students how to model and execute their first simple ChainReaction strategy. The remainder of this session and the ChainReaction sessions on Day 2 were focused on the development and improvement of game strategies. At various intervals additional hints and suggestions helped the girls improve their strategies, which became quite sophisticated toward the end of the second day.

In the last ChainReaction session, on the third day, every group had to select and submit their final strategy. They entered a final tournament with all strategies playing against each other to determine the best one. The bar chart in Fig. 3 (lower right) shows the results of a tournament of nine strategies.

Serious Games: The Potential for Business Applications

The development of strategies and strategic behavior is most relevant to entrepreneurs or management teams that have a clear vision and mission for their business and are in the process of developing primary strategies. The development of a suite of strategies is an iterative process and involves circular thinking on the basis that optimal strategies will evolve gradually and be very interdependent. Accordingly, the best way to test strategies is to define, model, and test them. By applying the ConnectIT approach to simulate business-related strategies, entrepreneurs and managers can easily evaluate their strengths and weaknesses in strategic planning and behavior (Fig. 4). Additionally under virtual conditions, the learning process for the managers and entrepreneurs is risk-free and costless. Furthermore, the results of a chosen strategy can be analyzed systematically and improved over time.

A venture is most prone to failure during its first three or so years of operation – the so-called valley of death (Murphy and Edwards 2003). A key to getting through these early years is to

Modeling of Games and Game Strategies, Fig. 4 Serious games in Second Life: business processes and workflows act in the virtual 3D experience

avoid the obvious mistakes. Generally speaking, businesses fail for significant and substantial reasons which are often very evident to outsiders. Insiders often fail to see them because of their closeness, determination, and so on.

These issues were reflected by the project *Science in 3D* (Lattemann and Stieglitz 2008) at Potsdam University. It employed a virtual world environment, Second Life, as an innovative tool for knowledge transfer for train entrepreneurs in the sector of biosciences. virtual worlds enable, thanks to their 3D environment, new ways of communication, collaboration, and cooperation. The level of interactivity of virtual worlds is higher compared to other web applications based on plain text, voice, or 2D presentations. Users in virtual worlds are represented by 3D avatars.

In this 3D training camp, entrepreneurs had to go through the whole cycle of a venture including selling products to customers. As Second Life had at that time a quasi-real economy, with an own currency and about 100,000 visitors daily who may act as customers, entrepreneurs could easily test their venture skills. The central part of this educational experience was the phase of "managing, growing, and ending the new venture." This phase includes the development of entrepreneurial strategy in the field of generating and exploiting new entries, growth strategies, and strategies for accessing resources and so forth (Hisrich et al. 2005). The project intended to provide simulations of the development of these kinds of strategies by implementing a ConnectIT-like approach for these strategies on a 3D basis in the virtual world. The one-thing approach to business process modeling in the jABC (Margaria and Steffen 2009) made this transfer to the business context easy and intuitive.

Discussion/Evaluation

If schools are offering "computers," what is taught in schools varies widely and can include HTML/web design, ECDL, browsing the web, typing, Scratch, GIMP, as well as working with Microsoft Word and Excel.

Quick Start and High Productivity

Model-based games like ConnectIT and ChainReaction combine an easily accessible application domain with a quick sense of achievement: The first game strategies are typically ready to run after a 30-min introduction to the modeling tool. Students can then work on incrementally improving this starting strategy. They also get immediate feedback about the quality of the results by playing against their own strategies or by letting the strategy play against other computer players. As such, the XMDD-based approach to games provides a very motivational framework and a fun and lightweight learning-by-doing way to acquire and practice computational thinking skills.

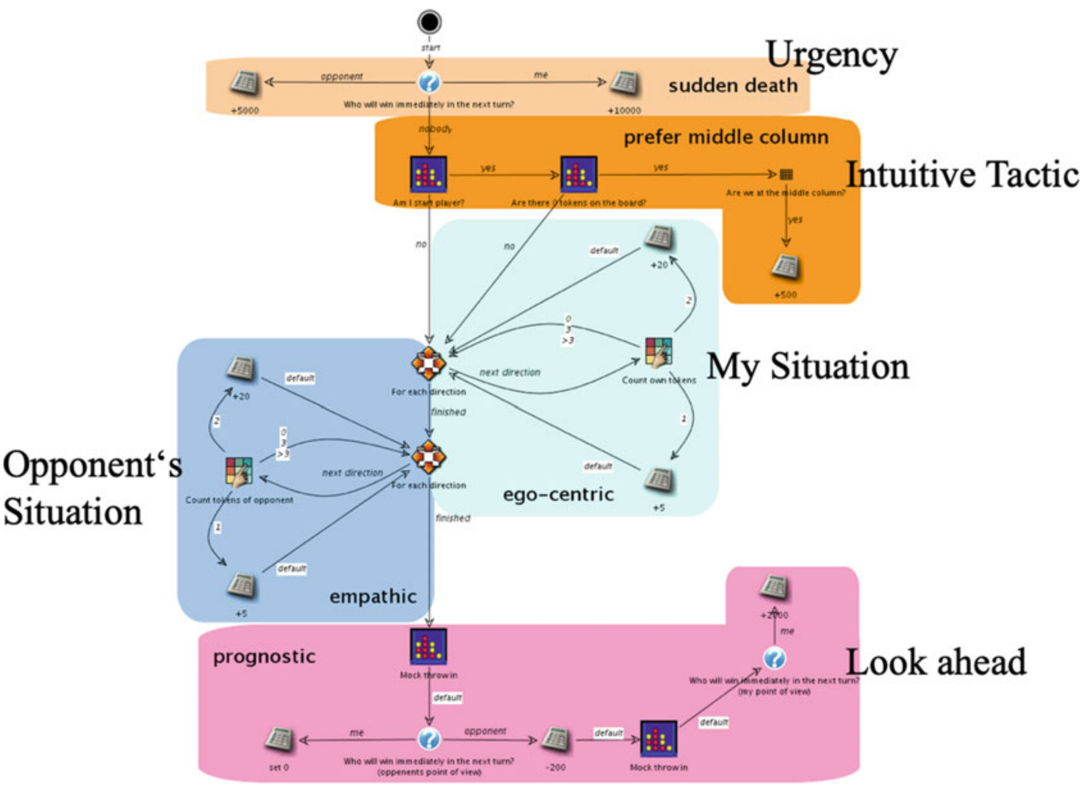
Note that the ability to be productive quickly is an essential trait mentioned when choosing scripting and interpreted programming languages like Python, PHP, or Lua over traditional programming languages like C, C++, or Java. The perceived simplicity of the language and the resulting improved confidence of the students are essential to avoid frustration due to struggle with complex environments.

In these applications of XMDD, the students end up mastering the essential concepts of a programming language through the domain-specific language of the game and are free to concentrate on the strategy design rather than on coding-level issues.

Ability to Express Complex Strategies

ConnectIT was used and refined in several rounds of the Do-Camp-Ing initiative (Do-Camp-Ing; ConnectIT) and in 1-day workshops (“Schnuppertage”) (http://connectit.cs.tu-dortmund.de/blog/), with heterogeneous groups of pupils who had no previous computer science training.

The pupils were able to build really competitive strategies like the one shown in Fig. 5 just by



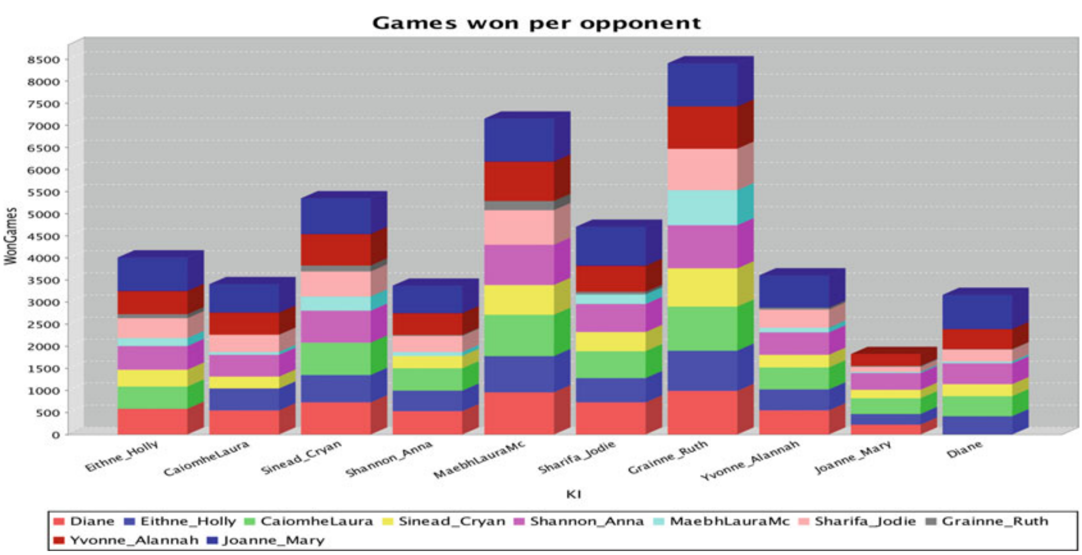
Modeling of Games and Game Strategies, Fig. 5 *Chaos AI*: A ComplexStrategy (by pupils) leveraging their intuition and knowledge about the connect-four game.

The *Chaos AI* strategy (name chosen by the two pupils who developed it) is very competitive. It contains the *Average Joe* component but adds a defensive part which considers the opponent's tokens. Furthermore, *sudden death* initially checks whether a throw in the current column leads to an immediate win. If so, it scores the column appropriately. Whenever there are no tokens on the board, the strategy *prefers the middle column*, a direct implementation of a rule of thumb (like the favored center position in the tic-tac-toe game) that emerges as a best practice when playing.

After counting the own and opponent's tokens on the board, the *prognostic* part of the strategy explores the near future via a mock throw-in at the current column. The mock throw-in building block exchanges the roles of the player and opponent; thus, all subsequent building blocks must be treated from the opponent's point of view (until another mock throw-in appears, as in our case). After the first mock throw-in, the strategy checks whether the opponent would win. In this case, one would help the opponent, so the score for the current column is 0. The other case is first punished with a moderate malus (−200 points), as one would build a too obvious triple that is likely to be blocked by the opponent. Second, another mock throw-in follows in this case (and the roles are switched again). If one would still win in this situation, the column gets a high score, as this indicates a catch-22.

Another approach for describing games in a concise manner was developed by the Stanford logic group and is termed game description language (Genesereth and Love 2005). Its main purpose is the description of games including the rules, the board, the players, and their interplay. However, the lisp-like notation of the language that in fact describes a state machine lacks in a graphical and understandable representation. For instance, a description of the connect-four game takes about 150 lines of code.

ChainReaction: The outcome of the tournament in the girls' camp is shown in Fig. 6. On average, the 24 strategies submitted to the final tournaments of the camps consisted of 19.16 building blocks (median 16, standard deviation 8.85) and had a McCabe number of 6.92 (median 7, standard deviation 4.13). The McCabe number (Murphy and Edwards 2003) gives the number of linearly independent paths through a process model and is commonly used quantitative complexity measure for software. That is,



Modeling of Games and Game Strategies, Fig. 6 ChainReaction: tournament results (from [WCCE]) although not being very large, the submitted strategies were by no means trivial and contained

various decision/branching points that create complex program flows, which is reflected by the McCabe number. With regard to readability and maintainability, typically a McCabe number of 10 is considered to be the upper acceptable limit of a program's complexity. The average McCabe number of the strategies is clearly below this threshold, indicating that they are indeed at a very comprehensible level.

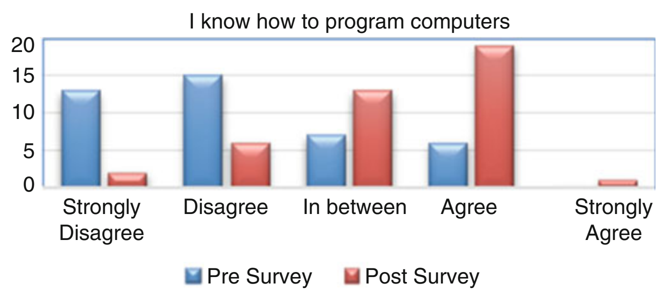
- In the **serious games** context, originally the most important aspect was the ability to model what-if scenarios, more than the 3D enactment of the scenes. The models were used to express and test hypotheses, to examine the consequences of decisions, and to validate properties and "rules" of the strategic decision-making under consideration. Two lessons were learned from this experience:
- The avatars and the simulation engine of Second Life were intriguing but not powerful enough to offer a realistic total immersion experience. Today's gaming engines are much better, and they are in fact supplanting dedicated simulators in a number of applications where augmented reality and an immersive experience play a role. A recent example is the remote action of a technician directing a local crew to perform maintenance operations on equipment.
- Modeling of the business decisions and strategies was useful and valuable also independently of the virtual world experience. In fact a new line of application of models was born from this experience, concerning the design and evaluation of business models (Boßelmann and Margaria 2014) using wizards to "direct" and guide the business user. Most recently this has evolved into a methodology for customization through knowledge sharing and configuration within enterprises (Steffen et al. 2016) and to systematically support decision-making through the hierarchical integration of heterogeneous business models (Steffen and Boßelmann 2018).

Impact of the Camps: Academic Exposure/Career Perception

In order to evaluate and assess the impact of our camps, we administered surveys at the beginning and at the end of the camps.

At the beginning students were asked if they had ever written a computer program. Fifty-one percent stated they had not written a computer program before, 26% of participants had written a computer program, and 23% did not know if they had written a computer program. As Fig. 7 shows, the participants' perceived ability to know how to program computers clearly increased between pre and post survey. Similarly, in our post survey, 95% of the students agreed and strongly agreed that they know more about computing as a job because of the camp. After the camp, we also asked the participants what they liked best about it and why, what they liked least and why, and what changes they would make to make it better.

One female participant in the mixed camps said: "I love the experience and the people even though we were split up in to groups and I was the only girl in the group I still made friends." Interestingly, when this student was asked about what she liked least about the mixed camps she said: "I didn't dislike most things the only thing



Modeling of Games and Game Strategies, Fig. 7 Survey results on "I know how to program computers" (ChainReaction)

I would change is putting at least one other girl into the group that I was in." This suggests that the young girls simply might feel uncomfortable being the only female in a group, and female-only or better balanced mixed-gender groups make them feel more comfortable.

Based on the participants' responses to the questionnaires, the girls-only camp was a very positive experience for them. Students' self-perception that they are good at computing and their perceived ability to know how to program increased by the end of the 3-day camp. In terms of the perception of computing as a career, participants knew more and were more positive about the field as a result of the camp. The answers to the survey questions also align well with the instructors' impression that the students had an enjoyable time while at the same time confirming that they learned a lot. The enthusiasm of the students can also be viewed in the video that was created from footage shot by the students on the last day of the camp (Google RISE).

The Google RISE funding award for the girls-only camp enabled specific actions to raise awareness about underrepresentation of females in particular.

Conclusions and Outlook

The development of specific offers for secondary school students has become a substantial part of universities' outreach and science communication activities in the last years. In order to be successful, they need to address attractive (or even fun) topics, which at the same time enable the students to learn something that is usually not part of their curricula.

A game-making project can significantly influence male and also female learners' attitude to computing (https://www.tandfonline.com/doi/full/10.1080/08993408.2013.774155).

A structured approach to training open-ended problem-solving can be of fundamental importance to nurture an independent but methodologically sound problem-solving (https://ieeexplore.ieee.org/document/7344088/), and the use of concrete tasks, for example, using LEGO Mindstorms, to engage students on algorithm design has proven successful for a long time (https://ieeexplore.ieee.org/document/6685052/). Games are more abstract, yet they typically make attractive course topics for students of all ages (IEEE 2014). They are also very suitable to address various aspects of software development, for instance, by looking more systematically at their mechanics, as we do with the development of strategies for computer players. The ConnectIT and ChainReaction workshops combine an easily accessible application domain with a quick sense of achievement: the first game strategies are typically ready to run after less than 30 min after the introduction to the jABC tool. Students can then work on incrementally improving this starting strategy. The students also get immediate feedback about the quality of the results by playing against their own strategies or by letting the strategy play against other computer players. Summarizing, it is a very motivational framework that provides a fun and lightweight way to acquire and practice computational thinking skills.

Concerning ConnectIT, this experience was extended to an interdisciplinary project in the project/seminar "Strategies in virtual worlds" for students and graduate students of CS, business and management, and biology at the University of Potsdam. There, ConnectIT was reinterpreted as a platform for strategy definitions and enactment in virtual worlds:

- Integrated in the Potsdam Second Life platform (Lehre in Virtuellen Welten)
- Applied as a 3D hands-on simulation environment of business, marketing, and communication strategies for founders of start-up companies – here mostly with a background in biology and life sciences, i.e., neither IT nor economy experts (Science in 3D – BMBF-Projekt)

Further editions and variations of the computing camp in the future also include a camp for secondary school teachers, training them to use the tools in their computing classes. Dedicated OER (open educational resources) artifacts in multiple languages are going to make it easier for third parties around the world to carry out similar courses. The software, documentation, and basic tutorials are already freely available (mostly in English, partly in German). Their systematic multilingual preparation and the development of additional material, like detailed lesson plans for school teachers or video tutorials in different languages, are ongoing. A first version was released into the Multimedia Education Resources for Learning and Online Teaching (MERLOT).

Cross-References

- ▶ Agile Methodology in Education of IT Students, Application of
- ▶ Computational Thinking
- ▶ eXtreme Model-Driven Development (XMDD) Technologies as a Hands-on Approach to Software Development Without Coding
- ▶ Modeling of Scientific Workflows
- ▶ Software Development Processes Designed for First Year Computing Undergraduates

Acknowledgments This work was supported in part with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero – the Irish Software Research Centre (www.lero.ie). We would like to acknowledge the support of Google RISE funding.

References

Allis V (1988) A knowledge-based approach of connect-four. Master's thesis

Bakera M, Joörges S, Margaria T (2009) Test your strategy: graphical construction of strategies for connect-four. In: Proceedings of the 2009 14th IEEE international conference on engineering of complex computer systems, ser. ICECCS '09. IEEE Computer Society, Washington, DC, pp 172–181. https://doi.org/10.1109/ICECCS.2009.51

Boßelmann S, Margaria T (2014) Domain-specific business modeling with the business model developer. In: ISoLA, Part 2, pp 545–560

Boßelmann S et al (2016) DIME: a programming-less modeling environment for web applications. In: Margaria T, Steffen B (eds) Leveraging applications of formal methods, verification and validation: discussion, dissemination, applications, ISoLA 2016. Lecture notes in computer science, vol 9953. Springer, Cham. https://doi.org/10.1007/978-3-319-47169-3_60

ConnectIT (2008) is documented and downloadable at. http://connectit.cs.uni-dortmund.de

Do-Camp-Ing (2008) http://connectit.cs.uni-dortmund.de/opencms/de/projekte/do-camp-ing-08.html

Genesereth M, Love N (2005) General game playing: game description language specification. Technical report March 15 2005, Stanford University

Google RISE girls' camp video (2015) https://www.youtube.com/watch?v=EzBXqouzd8k

Hisrich RD, Peters MP, Shepard DA (2005) Entrepreneurship, 6 edn. New York, United States: McGraw-Hill

IEEE (2014) Improving learning outcomes by designing engaging educational tools. https://ieeexplore.ieee.org/document/7044042/

IEEE (2016) A summer computing camp using ChainReaction and jABC. IEEE. https://doi.org/10.1109/COMPSAC.2016.41

jABC (2007) Website www.jabc.de

Juan AA, Loch B, Daradoumis T, Ventura S (2017) Games and simulation in higher education. Int J Educ Technol High Educ 14:37. https://doi.org/10.1186/s41239-017-0075-9

Lattemann C, Stieglitz S (2008) Einsatz Virtueller Welten in der Aus- und Weiterbildung – Das Projekt Bio-VWe. In: GENEME Gesellschaft fuür neue Medien, Dresden, Germany: TUDpress Verlag der Wissenschaften

Lehre in Virtuellen Welten – Projekt der Potsdam Graduate School (2008) http://www.uni-potsdam.de/db/jpcg/index.php?option=com_content&view=article&id=11&Itemid=20

Margaria T, Steffen B (2008) Agile IT: thinking in user-centric models. In: ISoLA'08, proceedings of 3rd international symposium on leveraging applications of formal methods, verification, and validation, Chalkidiki (GR), CCIS N. 017, Springer, pp 490–502

Margaria T, Steffen B (2009) Business process modelling in the jABC: the one-thing-approach. In: Cardoso J, van der Aalst W (eds) Handbook of research on business process modeling. Hershey, USA: IGI Global

Margaria T, Steffen B eXtreme Model-Driven Development (XMDD) technologies as a hands-on approach to software development without coding. EAIT, this volume, Springer. (please add the page numbers once they are known to the volume editors)

McInerney C, Lamprecht AL, Margaria T (2017) Computing camps for girls – a first-time experience at the University of Limerick. In: Tatnall A, Webb M (eds) Tomorrow's learning: involving everyone. Learning with and about technologies and computing, WCCE 2017. IFIP advances in information and communication technology, vol 515. Springer, Cham. https://doi.org/10.1007/978-3-319-74310-3_50

Murphy LM, Edwards PL (2003) Bridging the valley of death: transitioning from public to private sector financing. National Renewable Energy Laboratory, NREL/MP- 720-34036

Science in 3D – BMBF-Projekt (2008) http://www.uni-potsdam.de/db/biovwe/
Steffen B, Boßelmann S (2018) GOLD: global organization alignment and decision – towards the hierarchical integration of heterogeneous business models. In: ISoLA, Part 4, pp 504–527
Steffen B, Margaria T, Nagel R, Jörges S, Kubczak C (2006) Model-driven development with the jABC. In: Proceeding HVC'06, IBM Haifa verification conference, Haifa (Israel), LNCS 4383, Springer, pp 92–108. https://doi.org/10.1007/978-3-540-70889-67
Steffen B, Boßelmann S, Hessenkämper A (2016) Effective and efficient customization through lean trans-departmental configuration. In: ISoLA, Part 2, pp 757–773
Vlachopoulos D, Makri A (2017) The effect of games and simulations on higher education: a systematic literature review. Int J Educ Technol High Educ 14(1):1–33. https://doi.org/10.1186/s41239-017-0062-1