Computer Science Education
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/ncse20

From boring to scoring - a collaborative serious game for learning and practicing mathematical logic for computer science education

Andreas Schäfer [a] , Jan Holz [b] , Thiemo Leonhardt [b] , Ulrik Schroeder [b] , Philipp Brauner [a] & Martina Ziefle [a]

[a] Human-Computer-Interaction Center, RWTH Aachen University, Aachen, Germany

[b] Computer Science Department, Learning Technologies Research Group, RWTH Aachen University, Aachen, Germany

Version of record first published: 20 Apr 2013.

Computer Science Education, 2013
http://dx.doi.org/10.1080/08993408.2013.778040

From boring to scoring – a collaborative serious game for learning and practicing mathematical logic for computer science education

Andreas Schäfer[a], Jan Holz[b], Thiemo Leonhardt[b], Ulrik Schroeder[b]
Philipp Brauner[a]* and Martina Ziefle[a]

[a]*Human–Computer-Interaction Center, RWTH Aachen University, Aachen, Germany;* [b]*Computer Science Department, Learning Technologies Research Group, RWTH Aachen University, Aachen, Germany*

(*Received 11 July 2012; final version received 1 December 2012*)

In this study, we address the problem of low retention and high dropout rates of computer science university students in early semesters of the studies. Complex and high abstract mathematical learning materials have been identified as one reason for the dropout rate. In order to support the understanding and practicing of core mathematical concepts, we developed a game-based multitouch learning environment in which the need for a suitable learning environment for mathematical logic was combined with the ability to train cooperation and collaboration in a learning scenario. As application domain, the field of mathematical logic had been chosen. The development process was accomplished along three steps: First, ethnographic interviews were run with 12 students of computer science revealing typical problems with mathematical logic. Second, a multitouch learning environment was developed. The game consists of multiple learning and playing modes in which teams of students can collaborate or compete against each other. Finally, a twofold evaluation of the environment was carried out (user study and cognitive walk-through). Overall, the evaluation showed that the game environment was easy to use and rated as helpful: The chosen approach of a multiplayer game supporting competition, collaboration, and cooperation is perceived as motivating and "fun."

Keywords: mathematical logic; theoretical informatics; multitouch; game-based learning; serious gaming; collaboration; university level

1. Introduction

The increasing shortage of young academics in science and engineering in many countries is alarming, especially against the background that technological developments are an essential part of the welfare of graying societies. Even if the attraction of students to computer science as a basic

*Corresponding author. Email: philipp.brauner@rwth-aachen.de

discipline for technical developments is high in the beginning, computer science study programs at universities suffer from low retention and high dropout rates. In Germany, for example, college dropout rates oscillate between 32 and 37% (Heublein, Schmelzer, & Sommer, 2008), while the average dropout rate in other study programs is considerably lower (about 24%). Over 50% students of computer science switch to a different study program after three or four semesters. Even though these numbers are not easily comparable – due to different educational systems across countries – the National Science and Engineering Foundation report the number of students changing their major from computer science to a different field to be 20% (National Science Board, 2008).

According to students' own reports, the primary motive for changing the major computer science (or even leaving university without a degree) is the (too) high cognitive demand, especially in courses of theoretical informatics and mathematics. This finding is embarrassing at first, as the majority (70%) of computer science students have attended advanced mathematic courses in school (Heine, 2006; Heine, Spangenberg, & Willich, 2008). Even though there could be an imbalance between the (perceived) quality level of mathematical education at school and the requirements of math at university levels, insufficient mathematical education alone cannot account for success or failure in informatics. Other factors like motivation deficits and their causes also matter. Presumably, students' lack of motivation is also due to insufficient teaching formats that fail to overcome the high abstract knowledge in mathematics and do not combine the mathematical problem-solving approach with concrete, real-life phenomena.

To analyze the mathematical barriers in learning computer science at universities, we conducted interviews with 12 students of computer science. Among these were students who just finished their first year, but also students who were working on their thesis or even recently graduated. The guided interviews focused specifically on difficult lectures in their university career. Interviewees invariably named mathematical and theoretical computer science lectures as most difficult. Eighty percent of them needed retake at least one exam multiple times. Asked where the difficulties came from, students stated that they had troubles to strictly apply mathematical methods and to precisely execute mathematical proofs on their own. Especially, the lecture "mathematical logic" was reported to be one of the biggest barriers to successfully getting a computer science degree.

In this study, we report on a software tutorial that was specifically developed to support students in understanding mathematical logic and to increase study motivation by implementing a collaborative gaming environment.

1.1. Support – mathematical logic as a "hard" example

The lecture on mathematical logic is typically taught within the first two years of computer science education at university level. In the course, students learn about several kinds of basic logics and how to apply certain formal methods within these logics which include both syntactic and semantic procedures to derive conclusions within the scope given by the logic (Burris, 1998).

For many students in the field of computer science, this lecture is one of the most challenging and difficult courses to pass. Apart from the contents themselves, one main factor for the difficulty is the written examination at the end, which in general consists of two kinds of tasks. The first kind requires students to understand a given formalized problem and to solve it by using formal proof methods introduced in the lecture. Here, the main difficulty lies in coming up with the right idea how to solve the problem and which proof method to apply. The second kind of task is to solve a given problem instance with a method that has been explicitly introduced in the lecture for that purpose. Here, the procedures are usually clear; however, the time pressure during the exam frequently causes students to make errors or to follow correct but nontrivial paths which consequently provoke mistakes. When students talk about their experiences with exams, they often mention time pressure and the resulting nervousness as problematic because these factors influence their openness for ideas as well as their susceptibility to errors.

One possible course of action would be to train the students' ability to come up with the correct ideas when they are needed and, additionally, to train students to handle performance and time pressure, but this is usually beyond the scope of a conventional computer science study program and, in addition, is not focused on contents. Our approach, in contrast, seeks to train the students in executing the more schematic tasks in a way that they are able to perform them correctly even under time pressure. Our assumption is that if this ability can be efficiently mediated, those exam tasks may be more likely to reduce the pressure level rather than add to it, which would free the students to think about the more difficult tasks.

Among lecture topics, the resolution in propositional logic is the most relevant, and therefore, a perfect candidate to use as an example. The general concept of the resolution method reoccurs throughout the course for different and more complex logics than propositional logic. The resolution is a formal method to test whether a propositional formula is unsatisfiable, which also means that its negation is universally valid (Robinson, 1965).

The single valid resolution rule infers a new clause, if two clauses contain one complementary literal (one is the negative of the other). The empty clause is derived, if two clauses with exactly one literal in positive and negative form are present. If the empty clause, which cannot be satisfied, is derived, the unsatisfiability of the entire formula is shown. To get a complete and sound algorithm for deciding which assignments of variables fit a propositional formula, the resolution rules have to be combined with a complete search algorithm.

1.2. Motivation-games in computer science

While regular university courses might tend to fulfill the cliché of boring learning environments without fun or relation to the real life, there are a plethora of pedagogical approaches which aim at making learning more interesting, more personally relevant, and more fun.

One popular approach is serious gaming, which combines learning with playing a game. Serious games have an educational purpose, including teaching, training, and informing its players, wrapped up in an entertaining environment (Brauner, Calero Valdez, Schroeder, & Ziefle, 2013; Michael & Chen, 2006). Depending on the game design, serious games offer great possibilities for motivating students, promoting collaborative learning, or arousing enthusiasm by using competitive elements (Hakulinen, 2011). One educational key benefit of the success of serious games, though, is that learners need to be actively involved during the game, no matter whether it is a single player game, a collaborative game, or a competitive game (Shabanah, Chen, Wechsler, Carr, & Wegman, 2010). As computer games are "popular," "interactive," and "competitive," "simplify assessment," and "utilize entertainment" (Shabanah et al., 2010), they can be regarded as a motivating and promising way for teaching complicated and abstract concepts.

2. Related work

2.1. Learning models

With the instructional design theory, Gagné provides a plan to purposefully pursue the formulated learning objectives within the learning process (Gagné, Wagner, Golas, & Keller, 1988). According to the "nine steps of instruction," learners' attention needs to be gained first, in order to activate the motivation to learn, and to stimulate the recall of prerequisite learning, then the stimulus material should be presented, providing learning guidance and eliciting the performance. A subsequent feedback, assessing the learners' performance and promoting retention and transfer, is advised to provide sustainable learning performance. Furthermore, the Felder-Silverman learning model (Felder & Silverman, 1988) helps us to design these steps with respect to the different learning styles: sequential or global, visual or verbal, active or reflective, and sensing or intuitive. While differentiating between diverse learning types to enable individual learning certainly has its limits within learning environments, which are designed for collaborative learning, the model helps to avoid neglecting any of these styles. One of the findings of Felder and Silverman was that most people comprehend graphical representations of certain information better than textual representations.

2.2. Learning motivation-based implementation criteria

Overall, three motivation-based criteria have been considered specifically in this context: intrinsic motivation, motivation mediated through collaboration and competition, as well as a hedonic component through the game environment, which may conform to the well-known Premack principle (Premack, 1959), according to which students may be more motivated to perform a less desirable activity (understanding abstract materials) in order to get a more desirable activity (collaborative game playing).

2.2.1. Intrinsic motivation

According to Malone, when trying to motivate students to play a learning game, there are two different categories of motivation: intrinsic and extrinsic (Malone, 1980). While extrinsic motivation is induced by external stimuli, like additional points for an exam, intrinsic motivation arises from the activity of playing the game itself. There is a strong positive correlation between a learning activity's intrinsic motivation and the activity's learning effect (Gagné et al., 1988). Malone describes several heuristics for designing serious games and focuses on possibilities to make these games attractive to play. Some characteristics are decisive for individual learning without a group: curiosity, challenge, control, and fantasy. The interaction when learning within a group is targeted by different aspects like collaboration, competition, and recognition (Hejdenberg, 2005). These aspects are considered when designing the different gaming modes.

Another source for motivation is the fact that the usage of modern media and technologies, such as multitouch tables, is generally considered far more motivating and thrilling than the usage of the old and long-known ones (Heidrich, Ziefle, Röcker, & Borchers, 2011).

2.2.2. Collaboration and competition

Concerning working and learning in groups, several slightly different definitions for collaboration and cooperation can be found, although both terms are often used synonymously. Cooperation describes a division of labor, whereas in collaboration the same task is performed by all group members together (Dillenbourg, 1999); though, with regard to multitouch environments, this distinction is typically not made. Khaled et al. use the term of collaborative learning for both scenarios (Khaled, Barr, Johnston, & Biddle, 2009; Thalemann, 2004). Subsequently, collaboration is used as the general concept of "working together," whereas cooperation specifically means situations where a division of labor takes place. The benefits of collaborative learning can be seen in the fact that learners are more involved in the topic (Straub, 2001). Also, learners process the learning contents more actively and benefit the most when they already know some basics. Additionally, learners are supported in joint critical thinking and become aware of their own thinking processes.

2.2.3. Algorithm visualization

The majority of algorithm visualization approaches deal with concepts that are typically introduced in the basic data structures and algorithms lecture at university (Shaffer, Cooper, & Edwards, 2007). Only a small fraction of the remaining minority deals with mathematical algorithms, so the algorithm visualization within a serious game about resolution in propositional logic tackles a relatively unexplored area. Yet visualization of a mathematical algorithm does not max out the potential for the learning process. The activities of the learners are of greater importance for the learning process than the content of the visualization itself (Hundhausen, 2002).

One approach that considers these aspects is "Algorithm Visualization using Serious Games" (Shabanah et al., 2010). With this concept, an algorithm is represented in four forms: (1) *as text*, with the steps of the algorithm described to get a basic idea of its mechanism, (2) *as a flowchart*, with the working flow of the algorithm statically visualized as a graph, (3) *as a game demonstration*, in which algorithms are dynamically displayed as a self-running animation, and finally (4) *as a game*, in which algorithms are represented in an interactive serious game that simulates their behavior and visualizes their influence on the data structure. During subsequent execution of these different algorithm representations, learners pass through three consecutive processes: the viewing process, the playing process, and the designing process. Subsequently, a creation process can follow in which the learner develops his or her own algorithm text, flowchart, demo, and game (Shabanah & Chen, 2009).

2.3. Multitouch tables in education

As multitouch tables inherently support collaborative work (Khaled et al., 2009), there are plenty of possibilities to use these benefits for learning processes within serious games. Nevertheless, it is important to foster the process of collaboration to promote its positive effects as learners do not necessarily interact with each other just because the environment provides the possibility for it (Kreijns, Kirschner, & Jochems, 2003). Successful cooperation within collaborative work can only be reached if communication and a sustainable mutual understanding are addressed. Joint information processing needs to be provided in order to manage the dialog between learners, giving them the possibility to achieve coordinated task management and a cooperative task division. This, in consequence, allows learners to build up an interpersonal relationship, a space of reciprocal interaction, learning motivation and, finally, an individualized task orientation. In the game context, competitive elements refer to situations within the game in which the player faces a certain necessity to act in a suitable way (Vorderer, Hartmann, & Klimmt, 2003). Social competition, on the other hand, refers to the competition between human opponents. It is assumed that social competition in the context of serious games also increases the learning motivation (Koster, 2005).

As multitouch technology provides a wide range of opportunities for graphical user interface interaction, it has become a significant topic of interest within recent years (Schöning, 2010), applicable in different fields, for example, common office work (Wigdor, Penn, Ryall, Esenther, & Shen, 2007), geographic visualization (Tran, Anslow, Marshall, Potanin, & De Roiste, 2011), concept mapping (Martinez Maldonado, Kay, & Yacef, 2010), or in the serious games context (Zyda, 2005).

So far, multitouch learning applications can be found primarily within the elementary school sector, for example, software for mathematics MEL-Vis (Tyng, Zaman, & Ahmad, 2011) or the Multitouch Education Table (George, De Araujo, Dorsey, McCrickard, & Wilson, 2011). But as multitouch tables encourage students to experiment more with a problem and its solution (Piper & Hollan, 2009), the technology can be regarded as suitable for learners of all ages, especially as the usage of tables as interactive objects additionally promotes collaborative working by naturally providing space for an ideal group size of four learners (Schneider et al., 2010). Yet computer science education is using multitouch tables quite seldom, even though the technological development of multitouch tables is a highly popular topic in computer science research. One successful application is a multitouch puzzle game for children in elementary schools. In this game, the children have to assemble and disassemble certain figures, such as humans or bicycles, in order to learn the principles of generalization, modularization, and hierarchization (Nabbi, Brauner, & Leonhardt, 2010; Nabbi, Leonhardt, Brauner, & Schroeder, 2011).

3. Concept

In this section, we describe the concept of the game-based learning environment on multitouch tables. The concept contains the following pillars:

(1) We picked the resolution in propositional logic as a field of research and implemented a serious game for multitouch tables as a proof-of-concept. Propositional logic was selected as most relevant

showcase of mathematical logic. Also, the general concept of the resolution method reoccurs throughout the course for different and more complex logics than propositional logic.

(2) The learning objectives of the serious game are twofold: First, learners should acquire, learn, and comprehend necessary foundations of the method. Second, learners should be enabled to execute the resolution method without mistakes. We decided to place the learners in an open learning environment where they can freely decide which learning objective they currently want to address and by which concept.

(3) A series of eight game modes have been set up, which can be used consecutively or in random order. The term "mode" is chosen to differentiate our approach from the term "level" frequently used in computer games. Levels usually have to be completed one after another, whereas in our case, the learner can decide to skip some of the modes or to reuse one mode multiple times in a row. The application is based on the scaffolding method as an instructional approach (Hogan & Pressley, 1997) and provides four learning modes and four practicing modes for processing the learned material. The first four modes use different instructional approaches to instruct the learner about the subject matter. The second set of modes uses several kinds of cooperation, collaboration, and competition to deepen the learning experience.

(4) The current game design supports multiple users at the same time and focuses on two teams working together or against each other. If more than two users want to use the game, they can form teams. Multitouch tables provide excellent opportunities to support multiple users at the same time. For example, in one of the game modes, users can simultaneously carry out the resolution method against each other. In this case, both users (or both teams) stand at opposing sides of a multitouch table. Thus, they can focus on the task that has to be carried out on their side, and in addition, they can observe how the opponent is currently performing. Comparably, multiple users can complement each other when they have to perform a shared task on the table.

An in-depth description of the eight implemented learning and practicing modes of the serious game for mathematical logic is presented in the next section.

4. Implementation

Figure 1 shows the overall structure of the learning game. It consists of two main sections (the first one is the group of learning modes, and the second

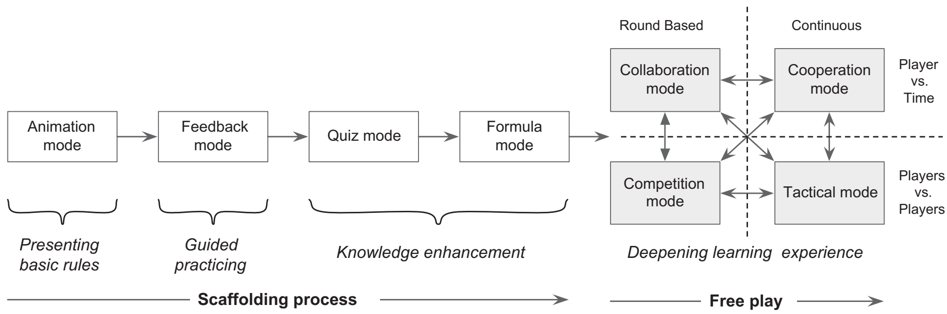

Figure 1. Schematic overview of the learning process and the gaming modes.

one is the group of playing modes) and one special mode for enhancement which is the creation mode.

In the general structure, two important construction ideas can be seen:

(1) Sequential composition: Beginning with the most basic learning mode that demonstrates the basic rules (animation mode), followed by a guided practicing mode (feedback mode), and modes offering knowledge enhancement (quiz mode, formula mode), and finally arriving at the playing modes, a learning sequence is traversed in which the contents and challenges of one mode are often based on the knowledge that has already been gained in another mode prior to the current one. The main goal of this sequential structure is to adequately guide learners through their learning processes without overburdening them by presenting them too much new information at once.

(2) Combination of building blocks: There are four different playing modes whose characteristics can be grouped into two categories. The first category is either a mode of purely collaborative nature (all players play together against time) or it is a competitive mode in which multiple players play against each other. The second category is either a mode made up of several subsequent rounds in which one single problem instance has to be solved per round, or it consists of one fluent sequence in which the number of problem instances that can be solved in a certain time is directly connected to the goal of the mode. The four modes now represent the four possible combinations of the respective two occurrences of both categories. The main goal of this combination strategy is to serve the preferences of different player types.

4.1. Application modes

The application consists of four learning modes that communicate important basic contents, and four playing modes to practice these contents.

The learning modes can be described as follows:

- Animation mode gives learners a first insight into how the algorithm works.
- Feedback mode allows learners to execute the algorithm themselves and delivers instant feedback.
- Quiz mode widens the learners' topical knowledge in the form of a quiz game.
- Formula mode enables learners to explore the algorithm's underlying concept (in this case, the construction of logical formulas).

The playing modes are designed to create intrinsic motivation so that players have fun while playing, and therefore, play the game because of the game itself and not because of pure need for practice. These modes can be described as follows:

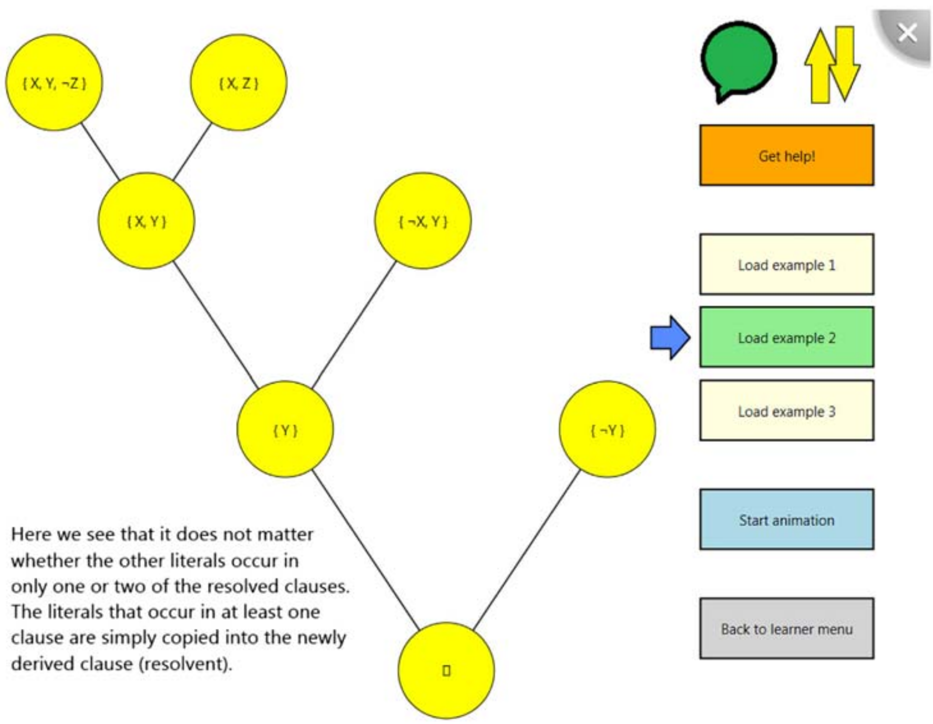

Figure 2. Screenshot of the animation mode: in this mode the execution of the resolution method is visually animated.

- Collaboration mode uses timers and round scores to allow players to play against time.
- Cooperation mode forces players to build teams with different responsibilities, playing against time, and toward one common score.
- Competition mode allows players to play a round-based duel against each other.
- Tactical mode provides a fluent and more advanced duel with a wider tactical range.

In the following, a short introduction of each mode's purpose is given.

Animation mode (see Figure 2) provides an initial demonstration of the algorithm execution which, according to Shabanah and Chen, is helpful to get learners into the topic (Shabanah & Chen, 2009). The chosen approach to demonstrate the algorithm behavior is comparable to what they call "algorithm game demonstration," an animation demonstrating how the algorithm works. The animation follows the same principles as the game later executed by the learners. In addition to the animation, an explaining text is shown when the animation is over. This text explicitly refers to the

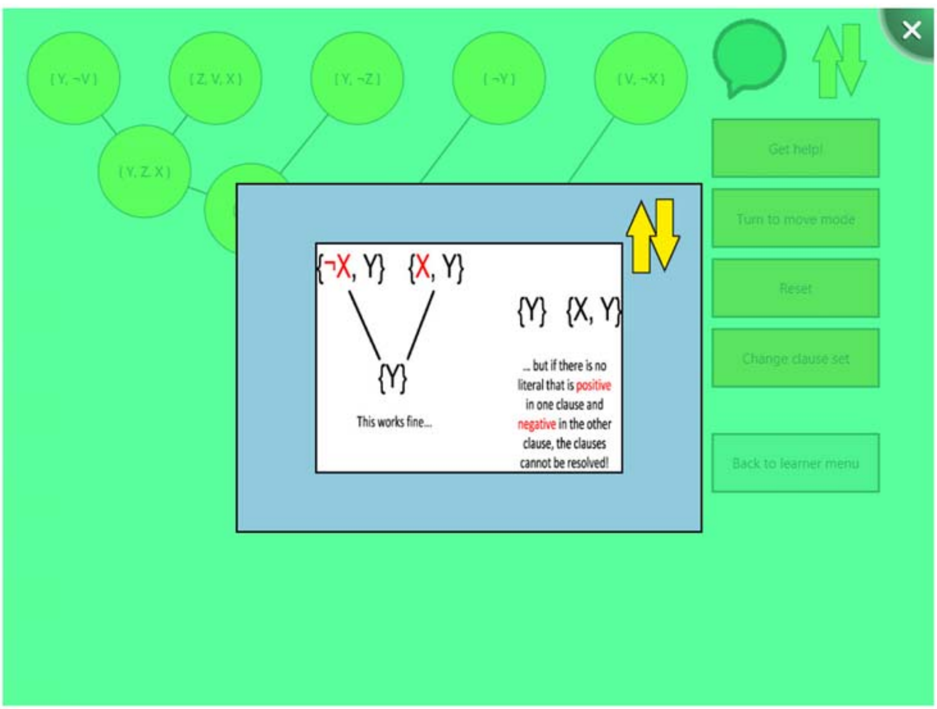

Figure 3. Screenshot of the feedback mode: in this mode, players can practice the execution of the resolution method and get immediate feedback for every action.

animation and aims at further explaining it and pointing to particularly important aspects.

Animation mode contains three different animations. The best way to teach multimedia-based lessons is to present them in user-paced segments rather than as a continuous unit (Pedroni, Meyer, & Oriol, 2009). Each animation contributes some specific information. The first one shows the most basic principles, while the other two each contribute additional information exceeding the information provided by the previous one. This step-by-step approach helps to avoid overburdening the learners with too much new information at once.

Feedback mode is designed to give learners direct feedback after every action to inform them whether their action was correct or, if it was not, what exactly their mistake was (see Figure 3). Continuous feedback supports learners to process the contents more effectively (Vollmeyer & Rheinberg, 2005), and thus, this application mode stimulates the learning process by giving feedback at every single step of the algorithm execution. If learners make a mistake, a special feedback popup appears. In addition to a short explanatory text, each popup contains several small pictures demonstrating the issue. For different kinds of mistakes, different messages appear, targeted at the type of mistake that has been made and explaining why it is a mistake.

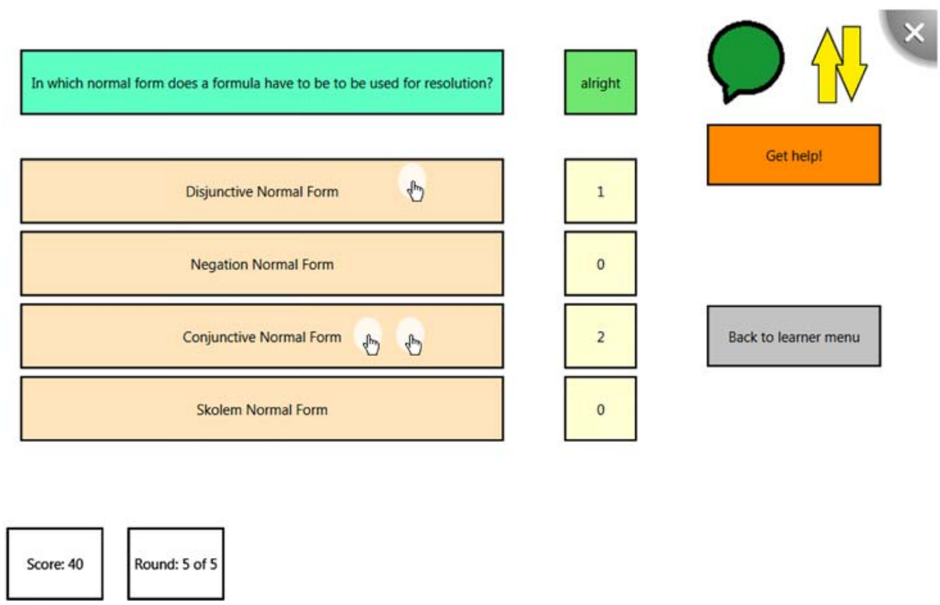

Figure 4. Screenshot of the quiz mode: players have to answer a series of questions. The majority of players decide which answer is chosen.

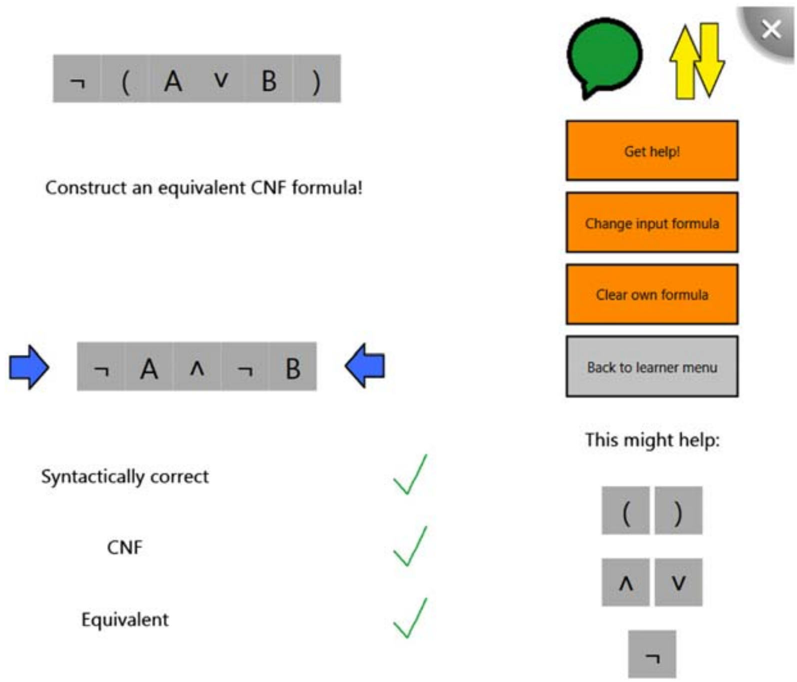

Figure 5. Screenshot of the formula mode: a given formula in propositional logic has to be transformed into a specific normal form.

Quiz mode provides several questions randomly chosen from a set of available questions that consider aspects having to do with the wider context of the algorithm (see Figure 4). This provides an understanding of the topic much deeper than the algorithm alone is able to communicate. The highest possible effect of collaborative learning is reached when some basics of the topic are already known but there is still a need to deepen and augment them (Koschmann, 1999). There are no time limits, so players have time to carefully think about the questions. When a group of players that have just learned the algorithm basics executes this mode, the players are led to discuss each question which can help to efficiently enhance the players' topic knowledge.

Formula mode is targeted at extending the main algorithm (resolution) by a prior step (formula construction). It gives learners the ability to explore which formulas are valid instances to be used in the resolution algorithm (they have to be in conjunctive normal form (CNF) and to construct an equivalent formula to a given formula (see Figure 5). The idea of formula equivalence checks is similar to the truth tables used in the truth tabulator (Van Benthem, Van Eijck, Van Ditmarsch, & Jaspars, 2011). In formula mode, learners can play around with basic formula building blocks and get direct feedback about syntactical correctness, CNF fulfillment, and

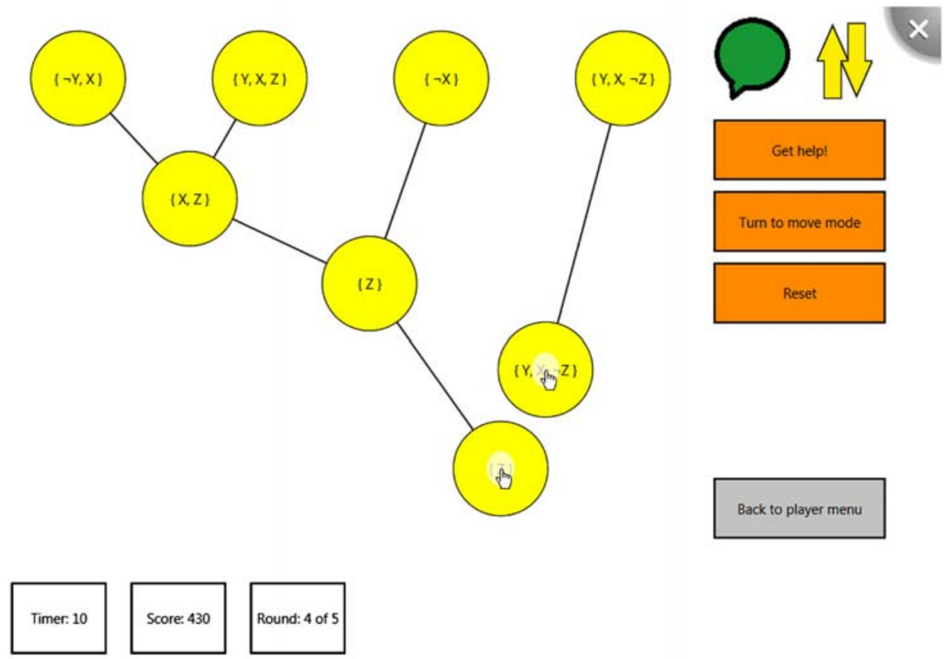

Figure 6. Screenshot of the collaboration mode: multiple players can simultaneously execute the resolution method.

equivalence. Similar to quiz mode, the purpose of this mode is to put the algorithm into a broader context.

Collaboration mode combines two central elements for creating intrinsic motivation, namely score-keeping and speedy response (see Figure 6). This happens in the following steps:

(1) Around starts. A timer is set to an initial value that depends on the currently chosen difficulty level. Each second the timer is decreased.
(2) Once the instance has been solved, the timer stops. The remaining timer value, multiplied by a constant factor, is then added to the score. If the timer has already expired, the players do not gain points for the round.
(3) If the round was the last one, the score is final. Otherwise, the next round follows and the round score is added to the score reached in the rounds before.
(4) If high score participation is activated and the score is high enough, it is added to the high score list which can then be viewed by the players.

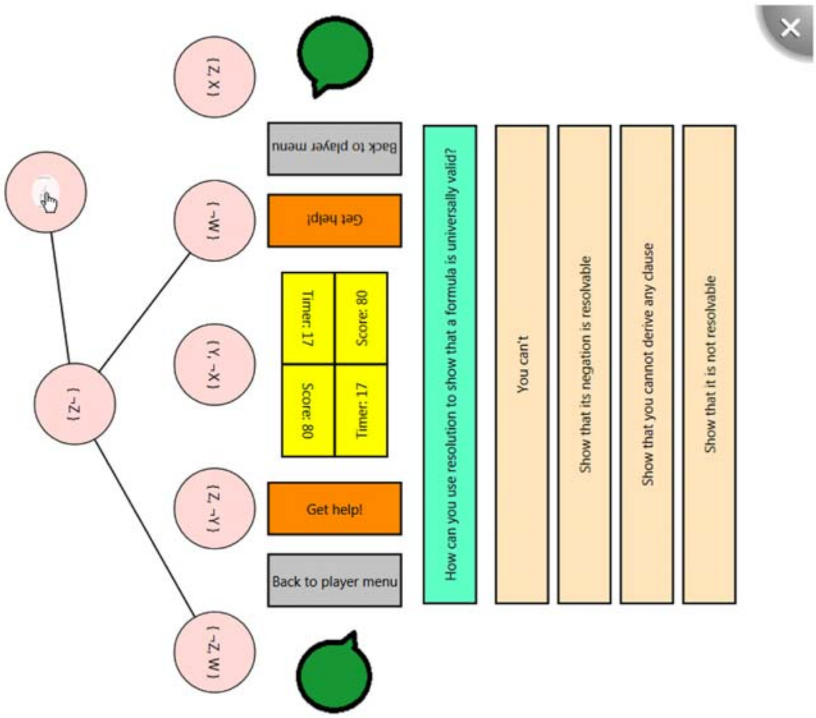

Figure 7. Screenshot of the cooperation mode: two users can still work together, but now they have different tasks to fulfill. One player has to execute the resolution method in limited time while the other player "buys" time by answering theoretical questions correctly.

The high score list can be classified as a first step toward competition because it causes players to compete against the already existing high score list entries.

Collaboration mode allows parallel interaction for multiple players. This ensures that several players are able to collaboratively work on the same instance at the same time so that every player, when having an idea how to proceed, is able to directly execute an action, regardless whether other players are executing other actions in parallel or not. In feedback mode, this kind of interaction is not possible because each action is directly followed by a feedback popup.

Cooperation mode (in contrast to collaboration mode, which is designed to support collaboration, but can also be played by one single player) explicitly addresses the aspect of cooperation (see Figure 7). As collaboration is suitable to get players more involved in the learning contents (Straub, 2001), making the collaborative aspect more explicit can be helpful and can be done by using the more special concept of cooperation.

In cooperative game situations, there are at least two separate roles players can take. Assuming two different roles, two different players are needed to fulfill these roles. If one of the roles stays unfulfilled, reasonable playing either gets impossible or is at least restricted. Multiple players playing together tend to intuitively understand and make use of the possibilities of cooperative game play (Khaled et al., 2009).

The two roles players can take can be described as follows:

(1) The first role is about executing the main algorithm. If an instance has been solved, the round does not stop but moves on and the score is increased by a constant value. Similar to collaboration mode, there is a timer running, but in this mode, when the timer has expired, the game is over and the reached score is final.

(2) The second role is about answering questions. Unlike in quiz mode, which is designed for knowledge accumulation, this quiz role aims at practicing knowledge recall under time pressure. When a question is answered correctly, a difficulty-level-dependent number of seconds is added to the timer value. When a wrong answer has been given, the timer value is analogously decreased.

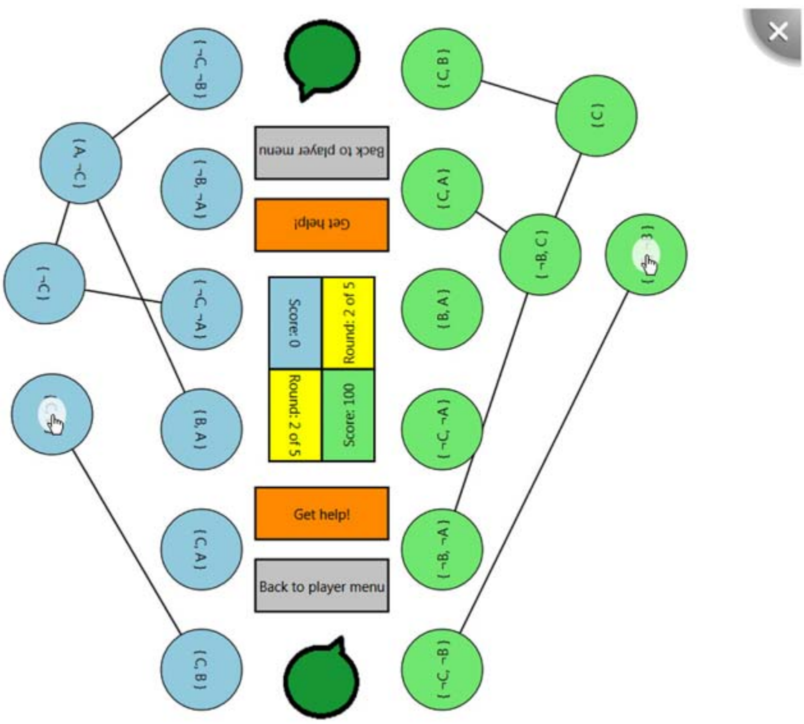

Figure 8. Screenshot of the competition mode: two players compete against each other on the same task. The player who finishes a round first gains points for the round.

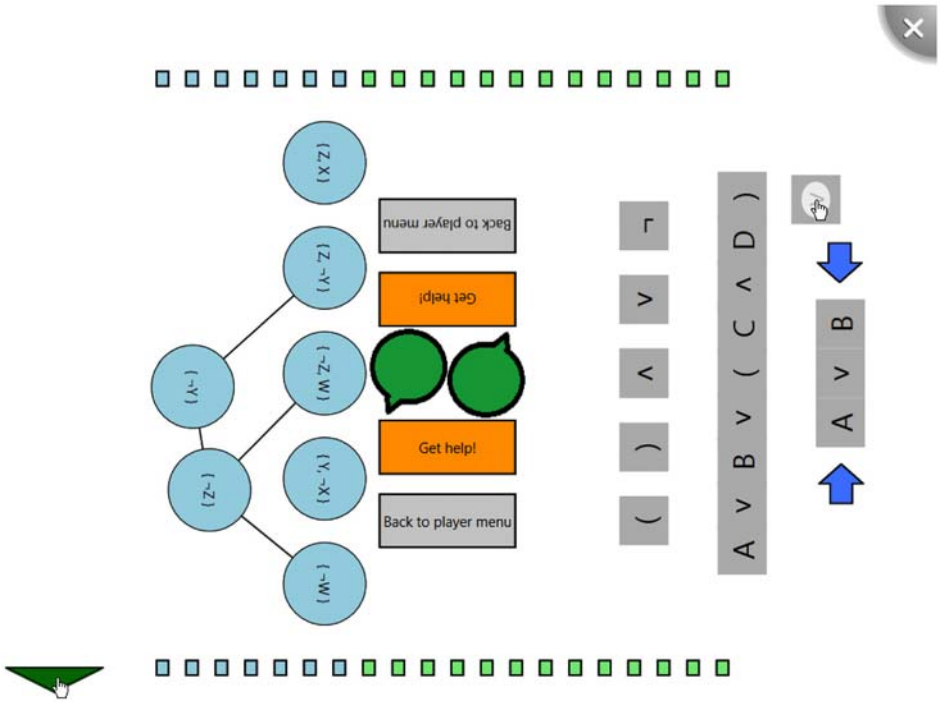

Figure 9. Screenshot of the tactical mode: in this mode players are competing against each other and can switch between different views like executing the resolution method or answering questions about the theoretical backgrounds.

Similar to collaboration mode, scores in combination with timers are the main challenging elements of this mode. Another aspect both modes have in common is the possibility to play for high scores to provide a basic type of competition.

Competition mode follows the principle of social competition between players facing each other (see Figure 8). In consecutive rounds, it gives two players exactly the same instances to be solved. There is no timer, but only the player finishing the round first gains score for it. At the end, the player who has gained more points during all rounds is the winner.

Tactical mode, similar to competition mode, offers two players the possibility to play against each other. The difference to competition mode is that, while in competition mode players can execute only one possible action at a time (solving one instance faster than the opponent), tactical mode adds the possibility to choose between different actions (see Figure 9). In tactical mode, there is a side bar (visible on both top and bottom side of the table), which represents the score difference between the players and consists of 20 squares. At game start, each player owns 10 squares, colored in the respective player's color. The game is over when all 20 squares are in one player's color.

The game starts with the view on a resolution instance. Players can now solve this instance. If a player solves one instance, the side bar progresses accordingly, changing the color of one square to the color of the solver.

Instead of solving the resolution instance, players can push an arrow on the right side of their view to switch to a quiz section. Correctly answered questions make players progress on the side bar, but with the quiz it is not possible to fill more than half of the side bar with squares of the own color. This is because answering one question can generally be seen as less time-consuming than solving one resolution instance. If the answer to a question was wrong, the opponent progresses on the side bar, independent of the current bar status. At any time, players can move back and forth between the resolution view, the quiz view, and a third view which shows the following five buttons:

(1) The "double button" doubles the number of squares conquered at any time the side bar progresses.
(2) The "force button" forces the opponent to solve his current resolution instance or quiz question, so that any other action is blocked until the task has been solved.
(3) The "another button" can be used to exchange the current resolution instance if in the current game situation it appears to be too time-consuming to finish it.
(4) The "formula button" offers the opportunity to access an extra view in which a logical formula can be chosen that the opponent can be forced to transform to CNF before being able to perform any other game action.
(5) The "home button" directly changes the view back to the resolution instance.

While the first four buttons can only be used once by each player and disappear after they are used, the "home button" can be used perpetually. Tactical mode integrates the concept of choice as well as the concept of force. Having the choice what to do adds a greater freedom of decision at each point of time, which allows players to act tactically. Being forced to do a certain task removes this freedom temporarily so that players cannot always choose the easiest possible way to win the game.

Creation mode follows a different approach than the other game modes. Most of the other introduced modes deal with learning or practicing to solve one certain resolution instance. Such instances have in common that they are predefined for use within the application which implies that they are actually solvable. One other interesting experience for players would be to try to solve an instance and then come to the conclusion that it is not possible, but for the sake of fluent playing, it is not sensible to integrate this concept into the existing game modes. Another problem is that, if the number of instances from which the application can choose is finite, eventually the same instances that have already been played appear again.

Creation mode addresses both problems. Similar to the "designing process" described by Shabanah and Chen, creation mode allows players to create their own instances (Shabanah & Chen, 2009). If players simply create an arbitrary instance and then try to solve it, it is very likely that they ultimately realize that solving it is not possible.

If the new instance is actually solvable, creation mode allows users to save it as soon as they have shown that it is solvable. It is then added to a custom data file in which all created solvable instances are stored. Players can always switch from using default instances in the playing modes to using custom ones.

It is important that only solvable instances can be saved. Trying to solve unsolvable instances and recognizing that they are unsolvable is an interesting experience, but if these instances were used in the application's game modes, it would negatively affect the game flow.

Additionally, if an instance has already been saved, players cannot save the same instance again, because many identical instances hamper the novelty aspect.

Creation mode also allows players to browse existing instances. These clause sets can then be used as starting points for creating more complex instances.

5. Evaluation

The implemented prototype of the serious game for learning and practicing mathematical logic was evaluated. In order to critically test the developed environment, we pursued two different evaluation approaches. First, a user-based evaluation was carried out. University students, who were currently taking the course in mathematical logic or had already finished it, were selected to use the prototype for learning and practicing the method of resolution in propositional logic. Second, an expert-based evaluation using the cognitive walk-through method (Courage & Baxter, 2005) was performed.

The user study was carried out in one of the university's seminar rooms. In total, ten participants (aged between 24 and 31 years) took part. They were allocated to five groups of two students, respectively. Each team used the prototype for about 10–20 min. Two experimenters observed the students during the interaction with the prototype and noted interaction and collaboration. After finishing the experiment, students had to evaluate the approach and were asked what they liked and disliked about the game and what suggestions they had for further improvements.

Overall, the prototype was regarded as easy to use and highly useful. All participants confirmed that the serious game approach increased their motivation to get into abstract and difficult learning materials. All of the 10 participants tried out many of the learning and gaming modes, although it seemed to be troublesome for some users to have every mode accessible right from the start. For example, one team just dove into one of the gaming modes without knowing how to perform the required algorithm (attracted by the possibility to play). Thus, they randomly touched and dragged clauses of a formula until they eventually found the solution by chance. They later suggested that users should be forced to pass the learning modes first before being able to access the playing modes.

Additionally, a series of usability problems were mentioned. First, the multitouch table had minor difficulties tracking all user interactions correctly. Although some interactions had to be repeated, the game was still well playable. Second, in some gaming modes, the screen space was too limited to display all required game elements. Some users suggested reducing object/font sizes. Third, participants reported that they did not like reading the explaining texts at the beginning of a mode. They preferred an introductory animation or video instead of a dull text. Finally, some users criticized that the *tactical mode* might not come to an end if the players perform equally well. Hence, they suggested introducing a time limit after which the game eventually ends. Though, this suggestion has to be treated carefully as other users actually liked this specific concept.

In general, all users agreed that playing the game is highly motivating, and they fully enjoyed dealing with the subject mathematical logic in a game-based environment. They specifically stated that practicing the resolution method with the game was much more fun than practicing this method with pen and paper.

The user study was complemented by a cognitive walk-through of the learning game prototype by two experts, both with a computer science background. The first expert was from the usability and human factors domain, the second was an expert in secondary and tertiary education and computer science teaching methodology. Both experts carefully tried out the different modes and evaluated interface design, educational approach, and the hedonic fun component in combination with cognitively demanding learning materials. Minor usability issues not revealed in the user-based evaluation were discovered: As such, in some learning modes, players stand at opposing sides of the multitouch table, and in others, they must stand on the same side in order to read texts and formulas. A unified solution has to be developed that does not require learners to move along the multitouch table. Also, the accessibility of all modes right from the start is criticized by the experts as it provokes two barriers for learning: First, some learners might directly access the advanced learning and gaming modes that are overtaxing without adequate knowledge of the subject matter. Requiring them to (quickly) progress through the learning modes will equip the learners with the necessary knowledge to perform well in the subsequent modes. Second, some (computer) gamers are motivated by challenges such as unlocking levels. Thus, unlocking additional game modes after successfully reaching a certain learning outcome might also prove as an effective agency to foster sustainable learning.

6. Conclusion

The application prototype presented in this study successfully applies the concept of serious gaming to the domain of mathematical logic, and therefore to computer science education. The user study and the cognitive walkthrough revealed that this serious game is suitable for learning mathematical logic and practicing the execution of the associated algorithms collaboratively, cooperatively, and also competitively. It uses the method of scaffolding to introduce the subject matter to the learners. The learners can then self-determinedly interact with the learning objects in various gaming modes, each addressing different learning styles.

Currently, the learning game deals only with parts of the contents of the lecture "Mathematical Logic." Thus, further efforts have to be made to cover the complete lecture. As stated at the beginning, the subject matter is basically twofold: First, learning the theoretical foundations of the subject and second, being able to apply the theoretical foundations in the form of specific algorithms in short time during exams. Adding additional learning elements in the form of questions in various game modes is relatively straightforward as the application already uses a database for storing the question elements. Supporting other algorithms in the training and playing modes requires additional interaction, validation, and feedback mechanisms to be implemented. As the algorithms learned in the lecture are basically more complex variations of the resolution algorithm presented here, adding further learning material from the domain of mathematical logic easy to accomplish. Still, additional considerations have to be made with regard to structuring the content in chapters and subsections as the lecture does.

There are other possible improvements to the application that can help to make it more appealing in order to further enhance its intrinsic motivation. One possibility is to include a graphic design artist into the process, who can take care of the application's visual appearance. There may even be room for replacing the currently used formal mathematical symbols by alternative representations, which could help to make the contents more intuitive for learners. However, it is important to provide a link to the formally correct notations at some point so that learners are not only aware of the principles but also of the correct formalisms.

Additionally, the effect of using this application in parallel to the mathematical logic lecture must be formally evaluated. The user study and the expert review described above provide first evidence that the application might have a positive impact on learning motivation and learning outcome; however, this has to be proven in a valid study.

One interesting aspect to investigate is the influence of the multitouch surface's collaborative nature on the learners' motivation. Here, the question is whether there is a verifiable difference when executing the application on a multitouch surface compared with when it is executed on a generic computer interface. Additionally, in a prior step, a meaningful measure for user motivation has to be found.

The presented learning environment will serve as an optional tool for students at our university to support self-directed learning in a game-based environment. It will complement the currently used teaching and learning style of lectures and weekly assignments.

However, the contribution of this work reaches beyond universities where mathematical logic is thought in comparable breadth and depth. The basic principles of scaffolding and mixing collaborative, cooperative, and competitive elements applied in this game may serve as a framework for developing serious games for a magnitude of different domains. It should be noted that this learning environment successfully applies the above-mentioned concepts to a very abstract mathematical domain which is often neglected by pedagogical research. Thus, the presented work provides strong evidence that even learning very abstract contents can be made accessible to cooperative and collaborative learning games. The utilization of a multitouch tabletop proved to be an excellent tool to promote collaboration and cooperation while learning this abstract subject matter.

Acknowledgments

We would like to acknowledge the valuable contribution of Jan Holz who sadly passed away before this study was concluded. We not only lost a committed colleague, but also a dear friend.

We would like to thank the anonymous reviewers for their constructive comments and valuable feedback on an earlier version of this publication. Also, we would like to express our gratitude to the participants of the user study and Chantal Lidynia for proofreading this publication.

References

Brauner, P., Calero Valdez, A., Schroeder, U., & Ziefle, M. (2013). *Increase physical fitness and create health awareness through exergames and gamification – The role of individual factors, motivation and acceptance*. Paper submitted to International Conference In Computing and Informatics (South CHI 2013), Maribor.

Burris, S. N. (1998). *Logic for mathematics and computer science*. Upper Saddle River, NJ: Prentice Hall.

Courage, C., & Baxter, K. (2005). *Understanding your users: A practical guide to user requirements methods, tools, and techniques*. Burlington, MA: Morgan Kaufmann.

Dillenbourg, P. (1999). What do you mean by "collaborative learning"? In *Collaborative learning: Cognitive and computational approaches*. New York, NY: Elsevier Science.

Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles. *Engineering Education, 78*, 674–681.

Gagné, R. M., Wagner, W. W., Golas, K. C., & Keller, J. M. (1988). *Principles of Instructional Design*. Belmont, CA: Wadsworth.

George, J., De Araujo, E., Dorsey, D., McCrickard, D., & Wilson, G. (2011). Multitouch tables for collaborative object-based learning. Design, user experience, and usability. theory, methods, tools and practice. In *First International Conference, DUXU 2011, Held as part of HCI International 2011* (Vol. 6769, pp. 237–246). Orlando, FL: Lecture Notes in Computer Science. Retrieved from http://dx.doi.org/10.1007/978-3-642-21675-6_28

Hakulinen, L. (2011). Using serious games in computer science education. *Proceedings of the 11th Koli Calling International Conference on Computing Education Research* (pp. 83–88). New York, NY: ACM. doi:10.1145/2094131.2094147

Heidrich, F., Ziefle, M., Röcker, C., & Borchers, J. (2011). Interacting with smart walls: A multi-dimensional analysis of input technologies for augmented environments. *Proceedings of the 2nd Augmented Human International Conference* (pp. 1:1–1:8). New York, NY: ACM. doi:10.1145/1959826.1959827

Heine, C. (2006). *Ingenieur- und Naturwissenschaften: Traumfach oder Albtraum? Eine empirische Analyse der Studienfachwahl* [Engineering and natural sciences: Dream or bad dream? An empirical study of the choice of field of study] (1st ed). Baden-Baden: Nomos.

Heine, C., Spangenberg, H., & Willich, J. (2008). *Studienanfänger im Wintersemester 2007/08: Wege zum Studium, Studien- und Hochschulwahl, Situation bei Studienbeginn* [Freshmen in the winter term 2007/08: Pathways for studying, field of study and university choice. The situation at the beginning of the study]. Hannover: Hochschulinformationssystem.

Hejdenberg, A. (2005). *The psychology behind games*. gamasutra.com, Retrieved from http://www.gamasutra.com/view/feature/2289/the_psychology_behind_games.php, February 2013.

Heublein, U., Schmelzer, R., & Sommer, D. (2008). *Die Entwicklung der Studienabbruchquote an den deutschen Hochschulen: Ergebnisse einer Berechnung des Studienabbruchs auf der Basis des Absolventenjahrgangs 2006* [Development of the dropout rate at German universities. Results based on the high school graduates from 2006]. Hanover: HIS Hochschul-Informations-System GmbH. Retrieved from http://www.his.de/publikation/vortrag/index_html?reihe_nr=U964

Hogan, K. E., & Pressley, M. E. (1997). *Scaffolding student learning: Instructional approaches and issues*. Northampton, MA: Brookline Books.

Hundhausen, C. D. (2002). Integrating algorithm visualization technology into an undergraduate algorithms course: Ethnographic studies of a social constructivist approach. *Computers & Education, 39*, 237–260.

Khaled, R., Barr, P., Johnston, H., & Biddle, R. (2009). Let's clean up this mess: Exploring multi-touch collaborative play. *Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems*. Boston, MA: ACM. doi: http://doi.acm.org/10.1145/1520340.1520680

Koschmann, T. (1999). Computer support for collaboration and learning. *Journal of the Learning Sciences, 8*, 495–497.

Koster, R. (2005). *A theory of fun for game design*. Scottsdale, AZ: Paraglyph Press.

Kreijns, K., Kirschner, P. A., & Jochems, W. (2003). Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: A review of the research. *Computers in Human Behavior, 19*, 335–353.

Malone, T. W. (1980). What makes things fun to learn? Heuristics for designig instructional computer games. *SIGSMALL '80 Proceedings of the 3rd ACM SIGSMALL Symposium and the First SIGPC Symposium on Small Systems* (Vol. 162, pp. 162–169). Palo Alto, CA.

Martinez Maldonado, R., Kay, J., & Yacef, K. (2010). Collaborative concept mapping at the tabletop. In *ACM International Conference on Interactive Tabletops and Surfaces*. Saarbrücken: ACM.

Michael, D. R., & Chen, S. (2006). *Serious games: Games that educate, train and inform*. Mason, OH: Thomson Course Technology.

Nabbi, N., Brauner, P., & Leonhardt, T. (2010). Vermittlung fundamentaler Ideen der Informatik durch eine kollaborative Multitouch-Anwendung [A collaborative multi-touch application for teaching fundamental ideas of computer science]. *Workshop proceedings of the Conference Mensch & Computer 2010*. Duisburg.

Nabbi, N., Leonhardt, T., Brauner, P., & Schroeder, U. (2011). Kollaborative und altersgerechte Lernanwendung zur Vermittlung fundamentaler Ideen der Informatik [Collaborative and age-appropriate learning environment for teaching fundamental ideas of computer science]. In H. Rohland, A. Kienle, & S. Friedrich (Eds.), *Proceedings of the 9th DeLFI conference – Die 9. e-Learning Fachtagung Informatik. Gesellschaft für Informatik* (pp. 185–196). Bonn.

National Science Board. (2008). *Science and Engineering Indicators 2008*, (Vol. 1, NSB 08-01, Vol. 2, NSB 08-01A). Arlington, VA: National Science Foundation.

Pedroni, M., Meyer, B., & Oriol, M. (2009). *What do beginning CS majors know? Computer* (Technical report/ETH, Department of Computer Science, 631). Swiss Federal Institute of Technology. Retrieved from http://dx.doi.org/10.3929/ethz-a-006835722

Piper, A. M., & Hollan, J. D. (2009). Tabletop displays for small group study: Affordances of paper and digital materials. *Proceedings of the 27th international conference on human factors in computing systems* (pp. 1227–1236). Boston, MA.

Premack, D. (1959). Toward empirical behavior laws: I Positive reinforcement. *Psychological Review, 66*, 219–233.

Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *Journal of the ACM (JACM), 12*, 23–41.

Schneider, J., Derboven, J., Luyten, K., Vleugels, C., Bannier, S., De Roeck, D., & Verstraete, M. (2010). Multi-user multi-touch setups for collaborative learning in an educational setting. *Cooperative Design, Visualization and Engineering, 6240*, 181–188. Lecture Notes in Computer Science.

Schöning, J. (2010). Touch the future: The recent rise of multi-touch interaction. *Journal of the European Commission's Future and Emerging Technologies Proactive Initiative on Pervasive Adaptation, 1*(1), 1–2.

Shabanah, S., & Chen, J. X. (2009). Simplifying algorithm learning using serious games. *Proceedings of the 14th Western Canadian Conference on Computing Education*. Burnaby: ACM.

Shabanah, S., Chen, J. X., Wechsler, H., Carr, D., & Wegman, E. (2010). Designing computer games to teach algorithms. In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on* (pp. 1119–1126). Las Vegas, NV.

Shaffer, C. A., Cooper, M., & Edwards, S. H. (2007). Algorithm visualization: A report on the state of the field. *ACM SIGCSE Bulletin, 39*, 150–154.

Straub, D. (2001). *Ein kommunikationspsychologisches Modell kooperativen Lernens. Studien zu Interaktion und Wissenserwerb in computergestützten Lerngruppen* [A communication psychological model for cooperative learning. Studies of interactions and knowledge acquisition in computer-mediated learning groups] (Dissertation). Universität Tübingen, Germany.

Thalemann, S. (2004). *Die Rolle geteilten Wissens beim netzbasierten kollaborativen Problemlösen* [The role of shared knowledge in network-based collaborative problem-solving] (Dissertation). Albert-Ludwigs-Universität Freiburg i. Brsg., Germany.

Tran, H., Anslow, C., Marshall, S., Potanin, A., & De Roiste, M. (2011). Tool support for map interface design: Individual versus collaborative approach. In *Australasian User Interface Conference (AUIC2011), Conferences in Research and Practice in Information Technology (CRPIT)*, Perth.

Tyng, K., Zaman, H., & Ahmad, A. (2011). Visual application in multi-touch tabletop for mathematics learning: A preliminary study. *Visual Informatics: Sustaining Research and Innovations, 7066*, 319–328. Lecture Notes in Computer Science.

Van Benthem, J., Van Eijck, J., Van Ditmarsch, H., & Jaspars, J. (2011). Logic in Action, last viewed at 2012–06-23. Retrieved from http://www.LogicInAction.org/

Vollmeyer, R., & Rheinberg, F. (2005). A surprising effect of feedback on learning. *Learning and Instruction, 15*, 589–602.

Vorderer, P., Hartmann, T., & Klimmt, C. (2003). Explaining the enjoyment of playing video games: The role of competition. *Proceedings of the Second International Conference on Entertainment Computing* (pp. 1–9). Pittsburgh, PA: Carnegie Mellon University.
Wigdor, D., Penn, G., Ryall, K., Esenther, A., & Shen, C. (2007). Living with a tabletop: analysis and observations of long term office use of a multi-touch table. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer System* (pp. 60–67). doi:10.1109/TABLETOP.2007.33
Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer, 38*, 25–32.