THE PRIMEGAME RELOADED: FINDING THE RIGHT BALANCE BETWEEN COOPERATION AND COMPETITION IN UNDERGRADUATE COMPUTER SCIENCE CLASSES

Manfred Meyer

University of Applied Sciences Gelsenkirchen, Campus Bocholt,
Münsterstr. 265, D-46397 Bocholt (GERMANY)
manfred.meyer@fh-gelsenkirchen.de

Abstract

While the PrimeGame approach has primarily been used in order to combine various foundational software engineering skills such as numerical mathematics, algorithm design and programming which usually are being taught as disjunct courses in many undergraduate CS (computer science) programmes, important soft-skills like teamwork and cooperation were not addressed so far.

The PrimeGame is a simple two-player board game which, however, is not fully-enumerable and thus not to solve with pure computational power. In the first years of PrimeGame history, students were required to implement autonomous players, for the game, i.e. small Java programs, in an attempt to model their own ideas on how to play the game best and eventually beat the opponent. All submitted players did finally compete individually against each other in a tournament setting.

Using the game in class, we witnessed an increased student interest in the subject matter, as well as a lot of fun among participating students. As the PrimeGame approach has so far been used and evaluated in undergraduate CS courses not only in European but also in African environments, it also exposed different cultural attitudes towards competition-based learning: While competition seems to be a more important part of European culture, the motivational factor of individually competing with other classmates was limited in the African context.

Therefore, we proposed and investigated a different setting for the use of the PrimeGame which combines both competition and cooperation: Students are no longer individually competing with each other but competing as a team against other teams. A straightforward implementation of this idea would be to let student teams develop one player together which then competes with other group's players in the tournament. However, this would soon lead to all the well-known issues in teamwork on a single common target as the individual effort and contributions to the team result may vary widely. For the PrimeGame therefore we propose a different approach: Every student still needs to develop and submit his own player to the competition. However, they also form teams of 3 to 5 students and the final team ranking is based on the performance of the best player from each team. Thus, students are forced to work together in the design of their players and find technical means for making their individual players cooperate in order to support one of them to outperform the best players from the other teams. As each student still has to develop and submit his own but cooperative player, usually recognized drawbacks of group projects can be avoided. Students still take part in a competition which has been proven to increase motivation and activation, but they can be successful only as a team, so they need to cooperate within the team and will also need to support other team members for the sake of the team's success.

While on a technical level developing cooperative players requires some additional CS skills, e.g. agent programming and communication protocols, and will thus extend the original PrimeGame approach making it best suited for a second-year course, it also exhibits a much better balance between the two driving forces of cooperation and competition.

Keywords: Algorithms, competition, cooperation, teamwork, complimentary skills, computer science, cultural aspects, education, games, Java, PrimeGame.

1 BACKGROUND

In most undergraduate CS programmes, basic skills like mathematical modelling, algorithm design, programming and complexity considerations are usually taught as part of different courses ([1], [4], [7]). For the development of suitable solutions, these and other skills need to be combined. However,

such combination often only takes place in advanced courses, focusing on larger and more complex projects, which then have to be developed and evaluated as teamwork. Based on previous success in game-based learning as shown e.g. in [6], [10] or [11], this observation led to the design and use of an easy-to-understand but rich-enough two-player board game for which students are asked to develop computer implementations of their individual strategies which will eventually compete against each other in a final tournament.

The history of this so-called “PrimeGame” dates back to the year 2003 when we first used this game-based learning approach in first-year CS courses at the University of Applied Sciences Gelsenkirchen. In fact, the PrimeGame was introduced in class after very few weeks of introductory programming lectures.

While students were almost instantly able to develop and apply basic game strategies when playing the game “manually" in class between two student groups, this turned out to be more challenging for them on the PC. Based on fundamental data and control structures, as were previously introduced in theory lectures, students now had to formalize their respective strategies, abstract formulas and algorithms into executable program code. Nevertheless, the initial task of developing working players was simple enough to be managed by students in little time, thus providing almost instant positive feed-back and encouragement for further discussions and algorithm refinements.

By also providing a framework for testing player implementations and running competitions as described more detailed in [2], students became highly motivated to not only deliver formally correct players, but also to develop and improve their strategies in order to come up with the best performing player for the final PrimeGame Competition which still usually takes place the week before Christmas.

In 2007, the PrimeGame approach was then used for the first time as part of the second-year course in Object-Oriented Technologies (OOT) at the Polytechnic of Namibia. Since then the Primegame Competition is held biannually as Namibian Open 2009 and 2011 [9], respectively, also attracting participants from outside the hosting institutions.

Although experiences from Germany and Namibia regarding the learning outcomes were found to be quite different (for a detailed discussion see see [2] and [3]), in both cases – thanks to the meanwhile very powerful and illustrative Competition GUI – the final PrimeGame Competition is regarded as one of the semester’s highlights and continuously demonstrates that programming and the design of efficient algorithms can also be a highly entertaining endeavour.

2 THE PRIMEGAME

The PrimeGame is a two-player board game, based on a very simple set of rules which can informally be stated as follows:

1. The *board* contains all natural numbers from 1 to some limit *n* (board size may vary but be constant for the tournament). Both players’ scores are initially set to 0.
2. The *players* make alternating moves until all numbers have been taken off the board.
3. A *move* is selection of one of the remaining numbers on the board. When making the move, the score of the player will be increased by the chosen number while the score of the opponent will be increased by the sum of all remaining factors of the chosen number still being available on the board. Finally, the selected number as well as all its factors are taken off the board and the other player is to make the next move.
4. The *winner* is the player with the highest score at the end. For fairness, every one game should consist of two *matches*, giving each player the first move in exactly one match.

Box 1: The PrimeGame Rules

These rules shall be illustrated using a small example: Assume a board size of 20, thus in the beginning the board contains all natural numbers from 1 to 20 (inclusive):

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Box 2: The Initial PrimeGame Board for board size n=20

The player making the first move is now to select any number from the board. Obviously, some choices are more advantageous than others, e.g.:

- Selecting number 19 will increase the player's score by 19 while only adding 1 to the opponent's score. Thus the player's effective gain (score difference) is 18.
- Selecting number 20 instead would increase the player's score by 20. On the other hand, as the set of factors of 20 on the board is {1, 2, 4, 5, 10}, the opponent's score will be increased by 22 (the sum of all factors on the board). Thus the player's gain for this move is negative: -2.

Obviously, number 19 seems to be a much better choice than 20. Of course, choosing any other number (from 1 to 18) would result in a legal move as well. Yet, as can easily be seen none of them would result in an equal or a better gain (the so-called *direct advantage* of the move).

After having selected number 19, the updated board as presented to the opponent to make his move will be the following:

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18		20

Box 3: The PrimeGame Board after execution of the first move selecting number 19

Obviously, this sample match will terminate quickly after a theoretical maximum of 10 moves for each player. However, with increasing board size not only does the number of moves for each game increase, but also the complexity of the task to select a "good" number (i.e. the player's *strategy*).

2.1 PrimeGame Strategies

A number of trivial strategies can easily be defined – and possibly be used as reference strategies to evaluate the performance of students' players. Following, we are briefly introducing a selection of the most basic strategies, which have been used for evaluation purposes in the past:

- The *"Anxious Strategy"* always selects the smallest available number on the board thus leaving no factor for the opponent.
- The *"Greedy Strategy"* always selects the highest available number on the board not caring about factors at all. Selecting number 20 in the introductory example is an instance of applying the Greedy Strategy.
- The *"Prime Number Strategy"* selects the highest available prime number on the board. If there is no prime number left, it turns into the Greedy Strategy.
- The *"No Factors Left Strategy"* selects the highest available number not having any factors left on the board. A variant of this strategy applies the Prime Number Strategy for the first move and thus avoids selecting number 1 in the first move.
- The *"Best Advantage Strategy"* selects that number leading to the best direct advantage, i.e. the highest gain (maximization of the score difference).

Although the latter performs a local optimization, all of the above strategies have been proven not to be optimal for sufficiently large board sizes. More successful strategies require a more sophisticated analysis of the consequences of each possible move for later selections – for the opponent and the player itself. While such game-tree analysis can be fully done for small board sizes or situations with only a small set of numbers left, its applicability becomes drastically limited as the number of possible moves increases.

Currently, a number of different player implementations exist which all show a better performance than the Best Advantage Strategy. These for example focus not only on the current move but also take possible counter moves into account, thereby optimizing the score increment after both: the own plus the opponent's move (expecting a best direct advantage for the combination of both).

However, a thorough number-theoretic analysis has not yet been completed and is assumed to be a non-trivial problem. It is therefore still an open question if a general-purpose optimal strategy exists for arbitrary board sizes and set-ups, how it would look like, and whether it has a manageable run-time complexity. While an "optimal strategy" can obviously be found with a brute force approach, such is hardly appropriate for reasonable board sizes as it would easily violate given time limitations per move.

2.2 PrimeGame Variations

In order to continuously use the PrimeGame approach every year, it has been necessary to slightly modify the rules from time to time such that player implementations from previous competitions may not perform as well as last time - if they perform correctly at all.

Such PrimeGame variations may affect the *selection rules*, e.g. stating that the highest number can only be taken once it has become the only one left, the *rewarding rules*, e.g. stating that for each prime number being taken the opponent gets an extra bonus of 25% of the prime numbers value, or some *general rules*, e.g. allowing for a limited number of non-moves per match meaning that no number is taken at all.

For the 2010 Christmas Competition at the University of Applied Sciences Gelsenkirchen the modified rewarding rules as mentioned above have been in effect. These allowed player implementations from previous tournaments to still take part in the competition, however with limited performance as they could not make use of the additional rewarding rule.

2.3 Practical Implications

For fairness reasons the board size to be used for all matches throughout the tournament is set *after* all players have been submitted, e.g. by using a random-number generator. While of course the board size itself can be any natural number, sizes ranging between 50 and 500 have shown to yield best results for classroom and competition environments - partly depending on the computer hardware used. Such boards are generally a good compromise between (intended) game complexity (and thus challenge) for the number selection task, and a reasonable run-time behavior allowing an audience to keep track of a complete tournament of up to approximately 150 players in separate rounds.

As we currently employ a round-robin approach during competitions in order to ensure every player faces every possible opponent exactly twice, groups of approximately 20 to 25 players still need to be formed as to limit the steep combinatorial increase in the number of overall matches.

3 THE PRIMEGAME FRAMEWORK FOR RUNNING COMPETITIONS

Currently, the PrimeGame framework features a highly modular design. It has been developed by Jens Fendler (a more detailed description can be found in [2]) and is composed of three main components: First, the Player interface(s) - defining valid player objects throughout the framework - which students must implement their players against. Second, the core Engine which sets up the board for each game, invokes the player objects' methods to take turns, to enforce the current ruleset, and to notify other components of game progress. And third, the Graphical User Interface (GUI) component which allows an easy tournament operation as well as visualisation of ongoing games.

The core *Player interface* is kept very simple. It primarily consists of two methods to be implemented: the `makeMove()` method which receives the current board state as a set of integer numbers and must return a valid number representing the player's move. The second method, `getPlayerName()`, is used to query the visible name of the player in order to display it during games and in the ranking table. Additional interfaces, inheriting from the Player interface, add methods to retreive the student number, as well as group identifiers to allow for automated group-based competitions. Currently, the GUI component also utilizes the student number (if provided through the corresponding interface) to load an image of the player provided in a file with the student number as a filename. For students not yet familiar with the Java collection framework, a simple wrapper class around the Player interface has also been developed. This wrapper converts the set of integer objects passed to the `makeMove()` method into an array of primitive integer values, and requires a primitive integer value as its return value. This way students can implement their players with virtually no knowledge of OO principles or Java API specifics, and are enabled to more or less directly map pseudo-code algorithms into working player code.

The *Engine* is the central component of the PrimeGame framework. It contains the complete game logic and is responsible for creating clean boards, enforcing the current rule set, invoking the players' `makeMove()` methods, and tries to detect any cheating activity by the players as good as possible.

Among others, the *GUI component* provides different board visualisations, statistics, current scores and ranking tables. It further allows the operator to reduce game speed by requesting delays in the engine during two moves. This allows the audience to closely follow a game as well as commentators to announce intermediate results or upcoming games to the audience. While the main functionality of the GUI is to provide an overview of the current game between two competing players, as well as to display the competition's ranking table including each player's score, it also registers a set of board renderers which give a comprehensive view on the current board, numbers taken, and available score.

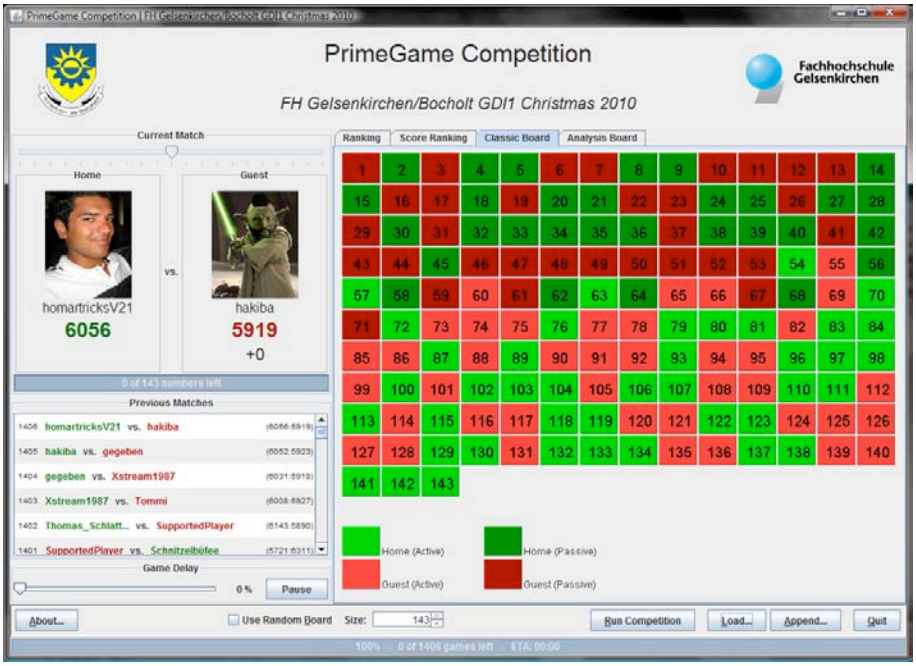

Fig. 1: The PrimeGame Competition GUI with “Classic Board” view

4 PREREQUISITES AND REQUIRED PRIOR LEARNING

In order for the PrimeGame to be successfully employed as a teaching tool, certain prerequisites must be met by the students. Clearly these requirements greatly differ depending on the intended learning outcomes, course level and the specific aspects of the game being studied. Yet we believe that the PrimeGame's potential is limited if used as one of the very first projects (i.e. early in the first semester). It was, however, successfully used as one of the final projects in first semester courses, effectively combining many of the skills students had acquired during their initial term.

As player development aspects will obviously be a primary aim in most CS teaching and learning scenarios, students should already have a fundamental understanding of certain key principles. First, a sound knowledge of program control structures, and foundational data types is obviously essential. Furthermore, basic abstraction skills are required in order to develop algorithms which will perform reasonably well on different board sizes, as well as against a variety of opponent players. As the PrimeGame - unlike traditional board games - leaves plenty of room for modifications (e.g. board size, number of participating players, etc.), students should try to develop as adaptive solutions as possible. Although the fundamental mathematics behind the game idea (or rules) is a rather trivial one, students will greatly benefit from a sound theoretical background in mathematics. Finally, apart from the ability to actually implement a given algorithm in e.g. the Java language, students should have a basic understanding of complexity theory and thus the run-time behavior of their solutions.

5 DIFFERENT VIEWS ON THE PRIMEGAME APPROACH

Over the past years, the PrimeGame has proven to be a powerful tool not only for supporting the learning process in introductory or basic programming courses. While originally being used in the very early stage of programming education, there is a huge variety of advanced aspects that can also be studied very effectively using the PrimeGame setting.

Depending on the course the PrimeGame is used in, the focus in an OO course may be more on interfaces, collection framework and using libraries, e.g. for game-tree search, while for the beginner's course wrappers are available that allow to simply access the board as plain integer array and thus fully concentrate on the core algorithm and the appropriate control structures. More advanced CS courses may find the PrimeGame being an interesting environment to study complexity issues or to focus on testing as both aspects can become critical when submitting a player to the tournament: Exceeding the given time limit per move or raising an exception while making a move will automatically disqualify the player.

6 EFFECT ON STUDENT MOTIVATION

Having used the PrimeGame as a teaching and learning tool in under-graduate CS courses for over six years, we have collected valuable experiences in different contexts. Although a through quantitative analysis of its impact is yet to be carried out, we have found some interesting qualitative results based on observations, project assessments, and feedback. Apart from the effectively measurable performance of the player implementations across different courses and student groups, we believe that student motivation is significantly determined by the game presentation. This assumption is based on project outcomes in comparison to the initial console-based to the current GUI front-end.

Overall, experiences at different institutions (University of Applied Sciences Gelsenkirchen in Bocholt and IT-Center at the International School of Management in Dortmund) show that when in practical classes it comes to introducing the PrimeGame and starting a two weeks period for player design, implementation and testing, most of the students almost instantly become highly motivated and keen to compete with other classmates. As usual, however, some students still choose not to participate as the PrimeGame is not a mandatory assessment due to formal reasons. Compared to other practical exercises, this number has been observed to be much smaller. Thus, the PrimeGame works very well for improving student's involvement and participation.

Unlike our expectations, however, differences in student participation, performance, and consequently player outcomes were observed between German and Namibian students. Contrary to some of the experiences in German universities, the achieved outcomes from previous PrimeGame projects at the Polytechnic of Namibia in southern Africa did not yield as promising results. It should be noted that the Namibian context is quite different from the German one in that it has profoundly different cultural settings, as well as an IT market often absorbing students regardless of their academic achievements. To some extend, this may result from underlying discrepancies in primary and secondary education as well as curricula content and delivery modes in these countries. Yet another important factor, the cultural context, is similarly likely to having contributed to the slightly weaker average performance of Namibian students as discussed in [2] and [3].

7 COMBINING COMPETITION AND COOPERATION

As it was especially learned from applying PrimeGame in the African context, competing with other students does not necessarily and always increase student motivation. In the European context it seems to work well, but on the other hand side important skills like cooperation and communication are not being trained using a purely competition-oriented approach.

Therefore, we proposed and investigated a different setting for the use of the PrimeGame which combines both competition and cooperation: Students are no longer individually competing with each other but competing as a cooperating team against other teams.

A straightforward implementation of this idea (*team player approach*) would be to let student teams develop one player together a group result which then competes with other group's players in the tournament. However, this would soon lead to all the well-known issues in teamwork on a single common target as the individual effort and contributions to the team result may vary widely. At the end, only one or two team members develop the player while other team members are trying to follow and understand what they are doing - in the best case! Thus, it can easily be observed that this approach will finally lead to a competition between the best heads from each team - leaving the other team members behind with limited contribution to the team result and thus limited motivation and learning at the end.

For the PrimeGame therefore we propose a different *cooperative players approach*: In contrast to the team player approach, every student still needs to develop and submit his own player to the competition. However, they also form teams of 3 to 5 students and the final team ranking is based on the performance of the best player from each team. Thus, students are forced to work together in the design of their players and find technical means for making their individual players cooperate in order to support one of them to outperform the best players from the other teams. As each student still has to develop and submit his own but cooperative player, usually recognized drawbacks of group projects can be avoided. Students still take part in a competition themselves which has been proven to increase motivation and activation, but they can be successful only as a team, so they need to cooperate within the team and will also need to support other team members for the sake of the team's success.

On a technical level developing cooperative players requires some additional CS skills, e.g. agent programming and communication protocols, and will thus extend the original PrimeGame approach making it best suited for a second-year course. Besides being a game for individual players, the PrimeGame can thus also be used very nicely to study the design and communication of cooperating agents: Instead of trying hard to implement the very best strategy, a group of students could try a different approach to win the tournament by “joining forces” and enabling their players to cooperate in a way that all but one of them support the remaining player. This can be done e.g. by applying a very poor strategy like the “Anxious Strategy” when it comes to play against a “friend” (the player to be supported) and to use a reasonably good one like the “Best Advantage Strategy” in all other cases. Here the focus will be more on defining an appropriate communication protocol as the only (legal) communication channel is the current set of numbers on the board. Such a communication protocol can easily be set up by defining characteristic first moves for the supported and its supporting or cooperative players. Once a cooperative player recognizes a previously defined characteristic move as answer to his special first move, the connection is established and the cooperative player will turn from normal to slave mode allowing the supported player to take as many points as possible from this match. Studies have shown that normally a small number of 3 or 4 supporting players is sufficient to outperform even the best available player provided that the ranking table is calculated based on the total sum of points earned from all matches performed.

Fig. 2 shows the final ranking table after a normal competition to which a team consisting of one supported player together with its cooperative players has been added. Here the ranking is computed based on the total number of points earned from all matches during the competition ("score ranking").

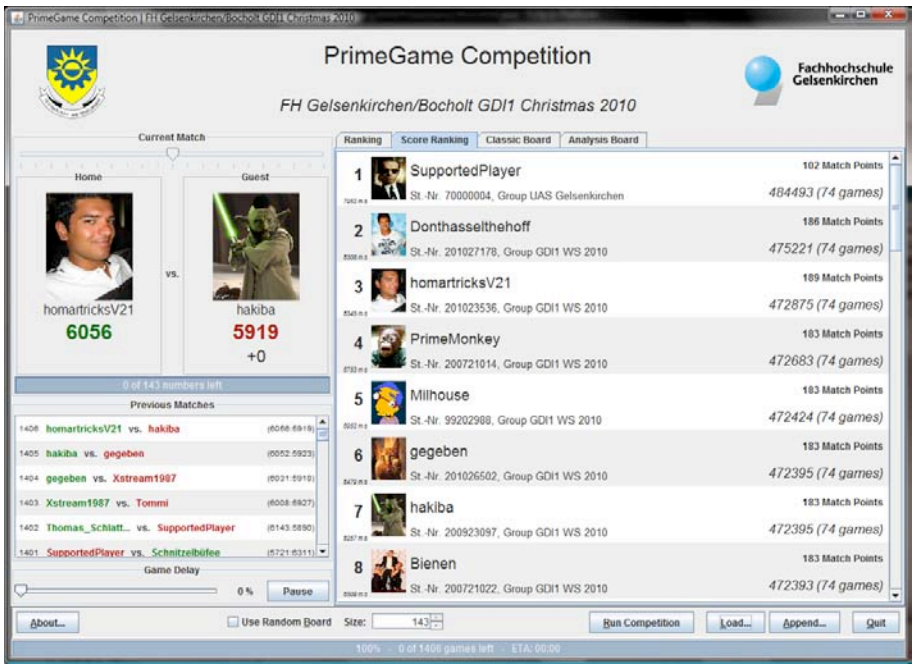

Fig. 2: Score ranking showing SupportedPlayer to win the competition

It shows the SupportedPlayer as the winner of the competition with a total Score of 484 493 out of 74 games. The second-placed got only 475 221 points out of the same number of games. However, the SupportedPlayer got only 102 MatchPoints (3 for each win) meaning that only 34 out of 74 games

were won while the remaining 40 games were lost. This shows that the games against its supporters provided enough points for the SupportedPlayer such that its implemented strategy need not to be very strong - and in fact appears not to be competitive with its toughest competitors in the direct matches.

Thus, for the same competition the alternative ranking based on MatchPoints (honoring victories more than the points earned, comparable to the modern soccer league ranking) does not even show the SupportedPlayer with the same Score and MatchPoints on the first top 8 positions (Fig. 3). In fact, this ranking lists the SupportedPlayer on position 21 only, followed by its supporters.

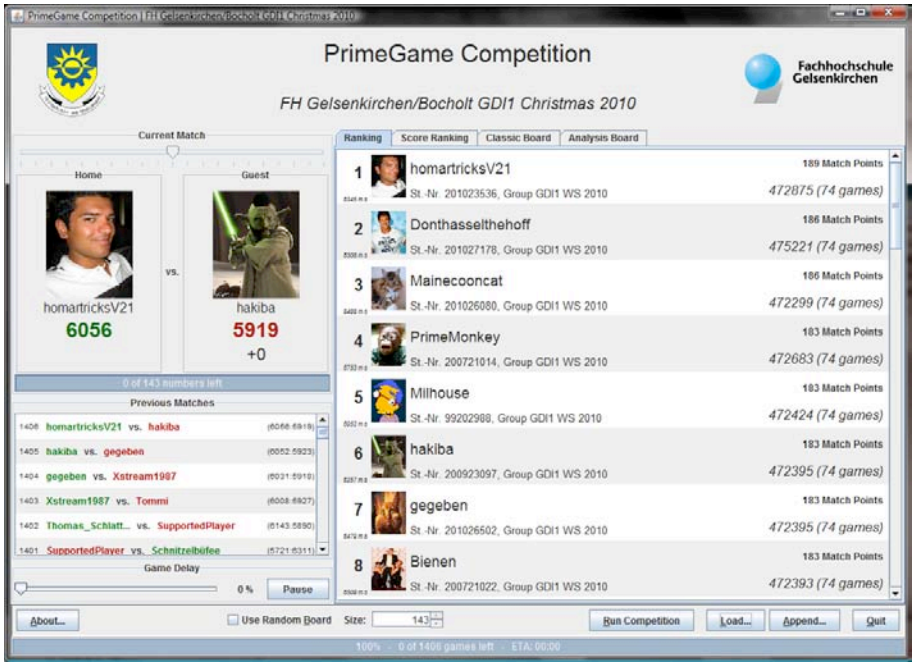

Fig. 3: "Traditional" ranking showing SupportedPlayer far from the top

This example illustrates very nicely that for the *cooperative players approach* it appears to be more important to develop a cooperative strategy within the student team rather than a strategy aiming at the best individual result. Thus, the focus moves towards the ability to recognize friends (more technically to define and implement proper communication protocols) and to support them in the best possible way (requiring the design and implementation of a strategy specifically tailored to match the strategy of the SupportedPlayer).

While still running a competition between entire teams which involves and thus motivates and activates all team members, other important aspects like teamwork, communication and cooperation skills are also trained making the PrimeGame a valuable tool not only for training computer science and programming skills in undergraduate classes.

8 FUTURE WORK

8.1 Player Algorithms

While we are currently using the PrimeGame mainly in undergraduate courses for Algorithm Design and Programming, far more advanced Player strategies could be developed in post-graduate CS courses. This obviously includes algorithms in artificial intelligence, but other areas such as Distributed Computing for the development of e.g. cooperative players also seem promising. Strategies based on large look-up tables or brute-force computations could similarly well be developed in cluster environments. Similarly, such approaches could incorporate networking aspects, resource constraints, or redundancy measures. The more theoretical aspects of PrimeGame players could furthermore be discussed in math-based courses on numerical methods or number theory to name just a few.

8.2 Online Framework

We are currently working towards an internet-based PrimeGame environment. While we would like to continue using as much as the original GUI and Engine code from the current framework, new interfaces and service layers between those components need to be introduced to allow for distributed competitions over TCP/IP networks.

A first working prototype has been developed already utilize the HTTP protocol with full proxy support (in order to tunnel the traffic through firewalls) as well as providing an own more efficient communication protocol [8]. The prototype allows users to register for a competition through a dedicated website, download the appropriate client framework (currently only available for Java but also planned for other languages) which then runs the player implementation locally and connects to our PrimeGame server to take part in either some fixed date competition or an ongoing tournament that runs forever. Especially for the latter sophisticated match-making and ranking algorithms have been developed already [5].

8.3 Teamwork and Intercultural Aspects

Having run PrimeGame competitions in two highly diverse environments - namely Germany and Namibia - already, it is planned to further expand the network of universities participating in PrimeGame competitions. While players from different places can easily compete in a common tournament, it might also be worthwhile to investigate intercultural teamwork aspects in distributed development projects. Based on our initial idea of providing the students with an interesting small-scale project early in their curricula, we believe this thought can very well be expanded to small-group projects in heterogeneous and spatially separated environments. Keeping such distributed project teams small, students might be able to more easily grasp group dynamics, and develop essential team skills in modern workgroup settings.

9 CONCLUSIONS

In this paper we have revisited the PrimeGame as a tool for game-based learning in undergraduate CS programs. We have outlined the game concept and motivated its use as a teaching tool in the first semesters, capable of successfully combining topics from a heterogeneous set of courses.

Necessary prerequisites in terms of prior student knowledge have been given, before having outlined our intended aims in employing the PrimeGame as a teaching and learning tool for applied programming.

While in a previous paper [2] we focused on the framework design and the results obtained from using the PrimeGame throughout several years including large-scale public competitions in Namibia which showed interesting differences compared to the European environment, the main focus for this paper was on the motivational goals that can be achieved using a new approach based on competition of teams rather than competing individuals.

This so called team player approach has been shown to result in a well-balanced combination of competition and cooperation. With this approach students are at the same time forced to cooperate on a common goal to which every team member contributes while the overall competition with other teams keeps motivation and activation high.

We have also presented a number of future opportunities around the PrimeGame concept and framework, which we hope can help us carrying over the previous success in few courses with increased student motivation and quality output into a greater variety of courses and projects.

Finally, we would like to point out the PrimeGame's potential as a collaboration tool among CS education institutions world-wide. Through involving lecturers and students at participating institutions simultaneously, and by allowing a public audience to follow entertaining events, we believe this game is a perfect vehicle to foster individual, institutional and professional partnerships. It is thus our hope that we can create further interest to participate in international PrimeGame competitions in the near future.

10 ACKNOWLEDGEMENTS

We would like to thank our students who have participated in the PrimeGame, and who not only came up with with a number of highly efficient and well-performing player algorithms, but also initiated very interesting discussions around different aspects of the game.

Arlo O'Keeffe, Oliver Krukow and Benjamin Boettcher contributed or still contribute to the future version of the PrimeGame (possibly "PrimeGame Revolutions") by developing the web-based architecture, match-making and ranking modules as well as admin front-end and database design.

Finally, very special thanks go to Jens Fendler at the Polytechnic of Namibia for uncountable discussions on PrimeGame, programming classes and so much more, for carrying forward the PrimeGame idea especially in Namibia, developing the competition framework and establishing the biannual PrimeGame Namibian Open in Namibia's academic calendar.

REFERENCES

[1] Fachhochschule Gelsenkirchen. Modulhandbuch (Syllabus) Bachelor of Science in Business Computing, http://www2.fh-gelsenkirchen.de/FH-Sites/fachbereiche/fileadmin/fb7/Downloads/Pruefungs-_und_Studienordnung/Modulhandbuch_BA_Wirtschaftsinformatik_09-07-07.pdf Last Accessed: 21 May 2011

[2] Fendler, J. & Meyer, M. (2010). The PrimeGame: Combining Skills in Undergraduate Computer Science Programmes. Proceedings INTED2010, Valencia, Spain, 2010, pp. 5454-5465

[3] Fendler, J. & Winschiers-Theophilus, H. (2010). Towards contextualised software engineering education: An african perspective. Proceedings ICSE2010, Cape Town, South Africa, 2010, pp. 599-607

[4] IT-Center Dortmund. IT Professional. http://www.itc-dortmund.de/it_professional_studieninhalt.php Last Accessed: 21 May 2011

[5] Krukow, O. (2010). "A Match Made in Heaven - Design and Implementation of Balanced Ranking and Matchmaking for the PrimeGame". Bachelor Thesis, Faculty of Economics, University of Applied Sciences Gelsenkirchen, August 2010

[6] Muratet, M., Torguet, P., Jessel, J.-P., & Viallet, F. (2009). Towards a serious game to help students learn computer programming. Int. J. Comput. Games Technol., 2009, 1-12.

[7] Office of the Registar (2010). "Prospectus for Undergraduate Studies". Polytechnic of Namibia, Windhoek, Namibia

[8] O'Keeffe, A. (2010). "Analysis, Evaluation & Optimization of Network Protocol Suites for turn-based Games - Demonstrated & implemented using the PrimeGame". Bachelor Thesis, Faculty of Economics, University of Applied Sciences Gelsenkirchen, August 2010

[9] Polytechnic of Namibia, School of Information Technology (2011). "PrimeGame 2011: 3rd Namibian Open". Online announcement on http://www.sit.polytechnic.edu.na/?q=node/188 Last Accessed: 21 May 2011

[10] Rankin, Y., Gooch, A., & Gooch B. (2008): The Impact of Game Design on Students' Interest in CS. In Proc. GDCSE Conference 2008. Cozumel, Mexico.

[11] Wang, A. I. & Wu, B. (2009). An application of a game development framework in higher education. International Journal of Computer Games Technology, 2009, 1—12