Using game theory and Competition-based Learning to stimulate student motivation and performance

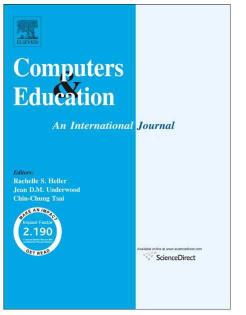
Article *in* Computers & Education · September 2010

DOI: 10.1016/j.compedu.2010.02.018 · Source: DBLP

1 author:

Juan Carlos Burguillo
University of Vigo

Accepted Manuscript



Title: Using Game-Theory and Competition-based Learning to Stimulate Student Motivation and Performance

Authors: Juan C. Burguillo

PII: S0360-1315(10)00052-7

DOI: 10.1016/j.compedu.2010.02.018

Reference: CAE 1585

To appear in: *Computers & Education*

Received Date: 22 November 2009

Revised Date: 16 February 2010

Accepted Date: 17 February 2010

Please cite this article as: Burguillo, J.C. Using Game-Theory and Competition-based Learning to Stimulate Student Motivation and Performance, Computers & Education (2010), doi: 10.1016/j.compedu.2010.02.018

Using Game-Theory and Competition-based Learning to Stimulate Student Motivation and Performance

Juan C. Burguillo
Department of Telematic Engineering
University of Vigo
36310-Vigo (Spain)
Email: J.C.Burguillo@det.uvigo.es
Phone / FAX : +34 986 81 38 69 / +34 986 81 21 13

Abstract

This paper introduces a framework for using Game Theory tournaments as a base to implement Competition-based Learning (CnBL), together with other classical learning techniques, to motivate the students and increase their learning performance. The paper also presents a description of the learning activities performed along the past ten years of a course where, in five of them, competition-based learning has been used. Finally, the experience gained is combined with an analysis of the feedback obtained from the students. The good survey results, and their similarity along the years, suggest that the combination of game theory with the use of friendly competitions provides a strong motivation for students; helping to increase their performance.

Keywords: Cooperative/Collaborative learning, improving classroom teaching, teaching/learning strategies, Competition-Based Learning.

Using Game-Theory and Competition-based Learning to Stimulate Student Motivation and Performance

Abstract

This paper introduces a framework for using Game Theory tournaments as a base to implement Competition-based Learning (CnBL), together with other classical learning techniques, to motivate the students and increase their learning performance. The paper also presents a description of the learning activities performed along the past ten years of a course where, in five of them, competition-based learning has been used. Finally, the experience gained is described together with an analysis of the feedback obtained from the students' surveys. The good survey results, and their similarity along the years, suggest that the combination of game theory with the use of friendly competitions provides a strong motivation for students; helping to increase their performance.

Keywords: Cooperative/Collaborative learning, improving classroom teaching, teaching/learning strategies, Competition-Based Learning.

1 Introduction

The use of games to promote student's learning has been done in the past to capture student's interest as all of us learn better when we are motivated (Bergin & Reilly, 2005). Most students have intimate contact with computer games before their formal computer education begins, and adequate computer games can attract and motivate them to learn more about computers in general, and programming in particular. Computer games as educational tools also have an intrinsic motivational factor that encourages curiosity (Kumar, 2000) and creates the impression to the students that they are in control of their own learning.

The use of games and competitions to promote the desire to improve has been explored in the recent literature in introductory data structures and programming courses (Adams, 1998; Becker, 2001). Other assignments include game projects (Huang, 2001), interprocess communication (Reese, 2000) and operating systems (Hill et al., 2003). Aside from the peculiar aptness of games for teaching AI, there are many pedagogical arguments to support the use of games in teaching in general, and programming in particular. In recreational computer games, players engage in processes such as proactive/anticipatory, recursive thinking, organisation of information, general search heuristics, means-ends analysis, and the generation of alternative solution paths (Pillay, 2002). In this approaches Game-based Learning (GBL) engages players in learning activities, usually by means of educational video or serious games. Serious games are designed to promote active participation and interaction as the centre of the experience, instead of pure entertainment. A few approaches, denoted as Competitive-based Learning or Competitive Programming, have successfully considered competitive games to promote learning in secondary and higher education, mainly in AI courses (Lawrence, R. 2004; Wallace & Margolis, 2007; Ebner & Holzinger, 2007; Ribeiro et al., 2009).

Game-based Learning (GBL) can be combined with similar learning methodologies as Collaborative-based Learning (CBL) (Slavin, 1980), Problem-based Learning (PBL) (Hmelo-Silver, 2004; Hmelo-Silver & Barrows, 2006; Merrill, 2007) and Project-based Learning (PjBL) (Barrows & Tamblyn, 1980; Boss, S., & Krauss, J. 2007). All these learning approaches are described next.

Collaborative-based Learning (CBL) methodology focuses on activities that maximize the collaboration among students, either in couples or small groups, to improve their learning activities and results. The idea is to enhance the exchange of information and knowledge among the students to motivate their own learning and a common reinforcement.

Problem-based learning (PBL) is a student-centred instructional strategy in which students collaboratively solve problems and reflect on their experiences. In PBL, learning is driven by providing open-ended problems where students usually work in small collaborative groups and they are encouraged to take the responsibility for organizing their group, and manage the learning process with specific support from a tutor or instructor that take the role of learning facilitator.

Project-based learning (PjBL) provides complex tasks based on challenging questions or problems that involve the students' problem solving, decision making, investigative skills, and reflection that also are supported also by a tutor that provides facilitation. The classroom projects are intended to bring a deep learning in issues related with their education. The difference between PBL and PjBL is that in the former (PBL) the teacher specifies the task to be performed at a basic granularity level, while in the latter (PjBL) the teacher specifies a greater task and let the students self-organize the subtask division. Previous experience in the field of Medicine also highlighted the usefulness of these approaches (Schmidt, 1983; Carlile et al., 1998; Morrison, 2004).

The idea of the work presented here is to combine all these learning approaches with game theory tournaments. Game Theory (Binmore, 1994) provides useful mathematical tools to understand the possible strategies that selfish individuals may follow when competing or collaborating in games. The context of cooperative games and cooperation evolution has been extensively studied in biological, social and ecological contexts, seeking general theoretical frameworks like the Prisoner's Dilemma (PD) (Axelrod, 1984). In his seminal work, Axelrod introduced a tournament among players, to obtain a winning strategy; which generated very interesting scientific results and a great interest for the whole scientific community.

The main contribution of this paper is the use of Game Theory tournaments to support Competition-based Learning (CnBL). CnBL is a methodology where learning is achieved through a competition, but the learning result is independent of the student's score in such competition; while Competitive-based Learning, or Competitive Programming, implies that learning depends on the result of the competition itself (Johnson & Johnson, 1985). CnBL can be easily combined with other learning methodologies as CBL, PBL or PjBL, and altogether they support tournaments among students' groups, used to motivate students and helping to improve their performance. We also compare the students' feedback and grades in courses where CnBL has been applied with others where it has not.

The rest of the paper is organized as follows. Section 2 provides a basic historical review of Game Theory. Section 3 describes basic information and the student profile for the course where the experience has been conducted. Section 4 presents the Game Theory approach for competition-based learning. Section 5 presents the learning activities done along the past ten years, and section 6 discusses the applicability of the CnBL approach. Finally, section 7 draws the conclusions.

2 Introduction to Game Theory

Game Theory (Binmore, 1994) provides useful mathematical tools to understand the possible strategies that individuals may follow when competing or collaborating in games. This branch of applied mathematics is used nowadays in the social sciences (mainly economics), biology, engineering, political science, international relations, computer science and philosophy.

Initially it was developed to analyze competitions in which one individual does better at another's expense: zero sum games (Morgenstern and von Neumann, 1947). From that moment, traditional applications of game theory attempt to find equilibria in these games. In any equilibrium each player of the game adopts a strategy that they are unlikely to change. Many equilibrium concepts have been developed; among them we find the famous Nash equilibrium, in an attempt to capture this idea (Nash, J. 1950). Game theory was developed extensively in the 1950s by many scholars and it was later explicitly applied to biology from the 1970s (Maynard Smith, 1982), although similar developments go back at least as far as the 1930s (Fisher, 1930).

The concept of cooperation evolution has been successfully studied using theoretical frameworks like the Prisoner's Dilemma (PD) (Axelrod, 1984). Axelrod has shown that cooperation can emerge in a society of individuals with selfish motivations.

2.1 Prisoner's Dilemma

In its classical form, the prisoner's dilemma (PD) is presented next and described in table 1. Two suspects are arrested by the police. The police have insufficient evidence for a conviction, and, having separated both prisoners, visit each of them to offer the same deal. If one testifies against the other and the other remains silent, the betrayer goes free and the silent accomplice receives a full 10-year sentence. If both remain silent, both prisoners are sentenced to only six months in jail for a minor charge. If each betrays the other, each receives a five-year sentence. Each prisoner must choose to betray the other (defect) or to remain silent (cooperate). Each one is assured that the other would not know about the betrayal before the end of the investigation. How should the prisoners act?

While this is the classical version of the PD game, it is presently more popular in its generalized form, which has been used frequently in experimental economics. There are two players and a banker. Each player holds a set of two cards, one printed with the word "Cooperate", the other printed with "Defect" (the standard terminology for the game). Each player puts one card face-down in front of the banker. By lying them face down, the possibility of a player knowing the other player's selection in advance is eliminated. At the end of the turn, the banker turns over both cards and gives out the payments accordingly. Table 2 provides the payoff matrix for this game where in every cell the first number stands for player A and the second number for player B.

The relations among the numbers in the payoff matrix are symbolized in table 3, where T stands for Temptation to defect, R for Reward for mutual cooperation, P for Punishment for mutual defection and S for Sucker's payoff. Depending on the relations among the different payoffs (R, T, S and P) we may have (4 ! = 24) different types of games, but to be defined as prisoner's dilemma, the following inequality must hold:

$$T > R > P > S \qquad (1)$$

This condition ensures that defection always pays more, but mutual cooperation beats mutual defection. In addition to the above condition, if the game is repeatedly played by two players, the following condition should be added to ensure that alternating does not pay enough:

$$2R > T + S \qquad (2)$$

In this last case, i.e., when the game is repeatedly played by two players, we have the iterated version of the Prisoner's Dilemma (IPD) which is described next.

2.2 Iterated Prisoner's Dilemma

In the iterated prisoner's dilemma (IPD), the game is played repeatedly. Thus each player has an opportunity to punish the other player for previous non-cooperative play. If the number of steps is known by both players in advance, economic theory says that the two players should defect again and again; no matter how many times the game is played (Binmore, 1994). Only when the players play an indefinite or random number of times cooperation can be an economic equilibrium. In this case, the incentive to defect can be overcome by the threat of punishment. When the game is infinitely repeated, cooperation may be a subgame perfect equilibrium (Axelrod, 1984).

2.3 Axelrod's Tournament

Axelrod's Tournament was played in the 80's, and initially he solicited strategies from other game theorists to compete in the tournament. Each strategy was paired with each other strategy for 200 iterations of an Iterated Prisoner's Dilemma game, and scored on the total points accumulated through the tournament. Even there were many complex strategies; the winner was the simplest deterministic strategy, of all the competing programs. It was submitted by Anatol Rapoport, and known as "TIT FOR TAT" (TFT) that "*cooperates on the first move, and subsequently echoes (reciprocates) what the other player did on the previous move*". The results of the first tournament were analyzed and published, and a second tournament held to see if anyone could find a better strategy, but TFT won again. Axelrod analyzed the results, and made some interesting discoveries about the nature of cooperation in these scenarios, which he describes in his book (Axelrod, 1984).

Although TFT is considered to be the most robust basic strategy, another interesting strategy is the one denoted as Pavlov (an example of Win-Stay, Lose-Switch) "*cooperates at the first iteration and whenever the player and co-player did the same thing at the previous iteration; Pavlov defects when the player and co-player did different things at the previous iteration*". For a certain range of parameters, and in the presence of noise, it was found that Pavlov beats all other strategies by giving preferential treatment to co-players which resemble Pavlov (Novak, M. & Sigmund, K., 1993).

3 Course Basics and Student Profile

The competition-based learning approach using game theory and the experimental data discussed in this paper has been performed along the last ten years in the course *Programming Paradigms*, which is an optional course for undergraduate students in Telecommunication Engineering at the Telecommunication Engineering School in the University of Vigo.

The course syllabus is mainly structured in three parts:

1) The first part introduces the history of different programming paradigms, and mainly describes: procedural programming, modular programming, object-based programming, object-oriented programming, logical programming and functional programming.

2) The second and more extended part is devoted to advanced programming in the Internet using Java. This part includes the use of Threads, Exceptions, Applets, Servlets, Serialization, graphical design (Swing), Packets, Interfaces, etc.

3) The last part is devoted to functional programming and exemplified with the Caml-Light language.

The course is organized into 15 hours for theoretical lectures (25% of the workload), 30 hours of practices and personal work (50% of the workload), 10 hours for tutoring and discussion (15% of the workload) and around 5 hours of using e-learning tools in the virtual campus (10% of the workload).

On average, students are 20 years old, and they have already taken courses on basic programming, computer architectures, software engineering and operating systems. Some of them have also followed courses on real time systems, distributed systems and computer networking.

To sum up, our average student can be characterized as follows:

- They want to learn more about advanced programming for the Internet in Java.
- They want to improve their competences in other programming paradigms not seen previously, mainly logical and functional programming.
- They want to follow a different approach in the programming classes, which concerns the competition-based learning oriented methodology described in this paper.

The course was first offered along the year 1996/1997 and the author has been the responsible of the course since then, but the data presented in this paper reflects only the last ten years (1999-2009), where the Competition-based Learning approach has been applied in five of them.

4 Game Theory to support Competition-based Learning

This section describes the practical methodology used for the lab work that has been applied to the students of the course. The lab work is centered on two practical jobs: one for Internet programming in Java and one for functional programming in Caml-Light. The former is three times longer (in time) than the latter, and it is the one selected for applying the game theory competition-based learning approach.

4.1 Lab Work Description

The first thing we do, at the beginning of the practical part of the course, is to divide the whole set of students in groups of two students each, that should work under a CBL approach between themselves. The students in the groups will keep together till the end of the course and every group shares the mark obtained for the lab work.

The lab work can be done at the lab or at home using the environments: Eclipse, Netbeans or the basic tools included in J2EE. There are no restrictions about where and when the students do the lab work (following a PjBL schema). Nevertheless, we have a general roadmap with milestones and every week; we provide at the lab a description of the task recommended at every milestone (PBL). This information is also available at the course web site in the Virtual Campus and it allows to follow a PBL, oriented to finish the whole coursework; providing orientation for lost students.

At the beginning of the course, we provide several documents to specify the tasks and to describe some topics related with the work to be done. Basically, we provide:

- A general description of the work to be done, and a requirements specification of the final tool that implements a particular game. Every group must submit its work, the game tool and a set of "intelligent" software players to play such game, before a certain deadline.
- A set of brief documents introducing concepts to be applied in the lab work: an introduction to Game Theory, an introduction to gaming techniques, an introduction to learning techniques from Machine Learning (Artificial Intelligence selected topics), an introduction to statistical techniques, etc.
- A final and real software executable (without the source code) to let them see how the tool and the players work. This reference tool is also useful, both for the students and the teacher, to test the interoperability between the main tool, developed by the teacher, with the players created by the students and vice versa, i.e.; the main tool developed by the students with the basic players created by the professor.
- A set of java files to be used as templates and or API files to be extended, while keeping interoperability, by every group. Table 4 presents an example of the *Player.java* file, where a basic player is defined to be later on inherited and rewritten by the students.

Therefore, besides a textual description of the job to be done, and a practical and fully functional implementation of the program, the students also receive a description of the API used for interaction among the players and the main program. The interface description takes into account the next topics:

a) The format of the file used to describe the players.

b) The directory structure to provide the game tool and the player files at the end of the course.

c) An example directory with the packages and contents to be used along the lab work, together with certain source files and classes describing the basic interface for developing the program.

The teachers guide the students along the project work, considering the roadmap defined and the students' evolution along the course, which are different every year. Nevertheless, students have complete freedom to select the techniques they will use for creating their "best" player.

4.2 Outcome

Every group must provide before the deadline:

1. A software gaming tool that implements the requirements specified where all the software techniques and course paradigms have to be used.
2. A set of basic players implementing some game techniques for testing the gaming tool.

3. A player per group (with a unique identifier) to be used in the final tournament.
4. A text file with a brief description of the tool, the players and the algorithm selected for participating in the tournament.

4.3 Tournament

The final tournament has been performed along the years at the final day of the course, when all the students do a short final exam to evaluate the theoretical knowledge they have learnt.

After the short exam finishes, which takes about one hour, we start the tournament among all groups and shown to the students by means of a beam projector. In this way, all the students see the public tournament and validate the results.

The first three players (i.e. groups) receive some extra points to be added to the mark obtained in the lab mark. Concretely, the first group receives two (over ten) extra points, the second group one and a half points and the third group one point. These numbers have the advantage that they have an equilibrated influence between 10 and 20% of the lab mark, i.e., they are an incentive. Besides, the groups that do not obtain any of the first three positions at the tournament do not suffer any type of penalty nor negative influence in their marks.

5 Results

In this section we describe the results obtained after applying the competition approach that has been described in previous section. This approach has been performed along five courses in the last ten years (2000-2009) and here we can see the feedback, the data and the statistical graphics for those years, together with some opinions from the surveys fulfilled by the students in the last three courses where the competition approach has been performed.

Next, it is described per every one of those three courses: an introduction to the lab work proposed, the algorithms provided by the groups that won the tournament, and the results of the survey answered by the students.

5.1 Course 2005: Matrix Game

In that course, we developed a game with a payoff matrix of 5 x 5 actions instead of the classical 2 x 2 actions for the Prisoner Dilemma. Initially, the payoff matrix is unknown by both players and they realize about its payoffs along the play. At the beginning of every game, player A chooses one of its actions (0, 1, 2, 3, 4) and the same is done by player B. Then, the main program proceeds to inform every one about the payment obtained by him and the opponent.

Considering N players, the final tournament consists in a (N x (N-1)) / 2 set of games or interactions between 2 different players (i.e., student groups) and every game between those two players is composed by 100 rounds (i.e., set of games between 2 players). The values of payoffs where selected randomly between 0 and 9 units.

Most of the algorithms submitted by the student groups have two main phases. An initial and transitory phase for exploring the payoff matrix, then a second phase is performed according with the strategy chosen by every group. The winner of the 2005 course tournament submitted a heuristic approach, strongly based on the experiences obtained playing against other groups prior the final tournament.

The students of this year fulfilled a survey at the end of the course, with the results shown in the table 5, where a value of 1 means is a poorest valuation and 5 the best one. As the reader may see, there is a clear tendency to support the competition learning approach based on topics related with Game Theory. The interaction with students from other groups is mainly neutral and strongly depends on each student attitude.

5.2 Course 2007: Peer-to-Peer Game

In that course, we developed a game, where we simulated a P2P model where N players participate exchanging files. The whole set of players is divided into G disjoint groups and every player belongs to one and only one of those groups. The number of groups (G) is configurable, and at the beginning, the players from the same group have the same set of files.

The aim of the game is to be the first to download all the files from the other groups, i.e., having the whole set of files in the P2P network. In order to model this, we do not use real files, but just use prisoner's dilemma matrixes to provide points through one to one interactions between players from different groups, until a minimum number of points per group are obtained.

To model the exchange of files, we define a 2x2 matrix where every player chooses between two actions: cooperate (C) or defeat (D). The defeat action models the behavior of users that do not share resources, but download resources from the others, and this problem is known as the Free-Riders problem or the Tragedy of Commons (Hardin, 1968). From the actions selected by both players the game tool determines the payment for each one.

The tournament is composed by a maximum of P games, where every game consists in a set of rounds between 2 players from different groups. It is not necessary to reach such maximum number of games if, at any moment, a player achieves the maximum score required for all the group files. At that moment, the tournament ends with a winner. The classification of the players is determined by the sum of points per group of files (limited by the maximum points in every file group).

The matrix used for this P2P game was the classical Prisoner's dilemma matrix of table 3. In every round, each player chooses its action (C or D) and both players inform the main program about their election. Then they obtain the action chosen by the other and the payment. Every player should store the amount of points obtained in each file group for organizing adequately its strategy.

The algorithms submitted by the students mainly include an alternated version of the TFT strategy. The winner of the tournament submitted an algorithm based in two phases: at the beginning it plays TFT in the four initial interactions to initialize a buffer of four actions. Then, it applies one of the next conditions:

- If there are two consecutive rounds where the opponent played C while it plays D, then it continues playing D.
- Else if there are two consecutive rounds where the opponent played C while it was playing C, then it plays TFT.
- Else, if the last action of the opponent was D, then it plays D, otherwise it keeps the current action just played.

The questionnaire fulfilled at the end of the course provided the results shown in the table 6. In this case, the practice proposed had a stronger complexity than the one done at the previous course and it caused a small reduction of the values for the third and fifth questions. The interaction with other groups has been also lower (2,79 over 5) than the previous year.

5.3 Course 2008: Guess-Coins Game

In that course, we developed a game that can be played by two or more players. At the beginning of every round, every player hides from zero to three coins. Therefore, the whole amount of coins hidden may vary among 0 and 3*N coins, where N is the number of players. Once all the players have hidden their coins, a starting player (who iteratively moves in a round-robin fashion) says how many coins he guesses there are, and then the next player tells its guess and so on, until the last player. Players can not repeat an election already taken by a previous player.

When the last player has done his election, then all the hidden coins are counted and the winner determined if there is one (there are more outcomes than players). On the one hand, if there is no winner the starting turn moves to the next player and the process is repeated. On the other hand, if there is a winner then it leaves the game for the next round. This process is repeated until only one player remains and loses the game. Thus, in every game with N players, there will be (N-1) winners and a loser. It is forbidden to fool other players about the coins hidden, for instance, a player can not say zero coins as his global guessing when he hides 1, 2 or 3 coins. These possibilities must be checked by the game tool. In this game a set of 2 to N players play a tournament as a set of repeated games, and every game is composed by a set of rounds where players leave the game until only one player remains and lost.

The algorithms provided by the students in this course mainly apply statistical approaches. The winner group applies the Big Numbers statistical law and considers every other player as a random variable. The result of combining all those variables is a Normal distribution, whose more probable value is the middle one between the minimum and the maximum. As the value obtained is a real number, they consider the lower integer value and if this value has already been chosen, then it increments and decrements the value until a free one is available.

The results of the questionnaire are similar to the ones of previous years and are shown in the table 7. It maintains the tendency to support the competition learning approach but the interaction with students from other groups increased a bit, probably because more players can play every game at the same time, i.e., it is not a set of interactions between 2 players as happened in the previous courses.

5.4 Qualitative Opinions and Evaluation Summary

In the surveys conducted in these three courses, besides numerical valuations, the students also have to provide qualitative answers related with the course experience. Next, a set of the most frequent student opinions is summarized:

a) Not too positive ones

- I do not like in general to compete. Nevertheless, to valuate with extra points the effort with respect to the others is good.
- I do not have any kind of competitive feeling, and it did not affect me in anyway.

b) Positive ones

- It is always good to have a bit of competition. It makes the task more interesting.
- It causes a rising of interest and makes us try to improve our competences.
- It is OK, when there is a reasonable limit in the weigh of the whole mark.
- I think it is positive to have some extra points, and it was really funny.
- I like it because it creates more interest.
- I think it is good to a certain level, if mainly, it helps to improve the final mark.
- It is interesting since you get more involved in the group and the practice.
- It makes more interesting the development of the practice.
- I liked the idea and it was funny to compete with other groups.
- The competition among groups enhances the relation with my student mates.
- Moderate competition stimulates learning and provides a playful approach.
- I like it, even my player did not win.
- I think that better ideas, that empirically win, must be valuated.
- It is positive because it increases the interest in the course and it motivates me more about the tasks to be done.

Considering the numerical results for the three courses altogether, we obtain the final results that appear in the table 8. The students consider that the topic proposed for the lab work has been interesting (4.16), the interaction with students out of the group has a middle value (3.02) but considering that in a normal lab work this result normally gets closer to one it is reasonable. Concerning the last three questions, the competition approach has been clearly an stimulus to learn more (3.86), the students strongly like the approach proposed (4.24), which is the highest value, and finally; the use of Game Theory tournaments is interesting to be applied in practical tasks (3.97).

Looking closer to the values obtained, we see that the results from question number 2 are much spread all over the different opinions. On the other hand, the rest of the questions are centered on the values from 3 to 5, being 5 the most popular value for the first (61 of 125) and the same happens with questions third (43) and fourth (67).

Table 9, shows the similarity of the average results obtained in the surveys along the years, which is relevant; considering that the lab work proposed have been relatively different. There is a small variation around 5% (except for the second sentence due to its higher variance). The similarity of the results described in table 9 validates the predictability of the surveys, and together with the textual opinions provided by the students suggest that the motivation is enhanced when playing friendly competitions. This enhancement normally helps to learn better and to improve the performance (Bergin & Reilly, 2005). From these survey results, we can mainly resume that the competition approach is strongly valuated by the students and that the use of Game Theory motivates them to do better their lab work.

In figure 1 we can see that the percentage of students passing the course is higher in the 5 years where the competition-based learning approach has been used (96,56% in average) than in the other 5 years without CnBL (90% in average). Besides, all the courses where CnBL has been applied have better results than all the courses when it has not, with an exception in the year 2002. Figure 2 shows the average mark obtained by the students along the courses. Here it is clear how the average mark obtained using competitions (8,6 over 10) is higher than the average mark obtained without it (7,8 over 10). We also see how the average marks in the years where CnBL was applied is higher than when it was not, only 2003 has a similar average mark than 2002 and 2001. Finally, table 10 summarizes the data used in this section. The previous figures and data suggest an improvement in the students' performance, in those courses where the CnBL approach has been used. Nevertheless, considering the novelty of these empirical results, it is important to remark here that it does not mean that CnBL is better than to any other learning technique. Instead, it points out about its interest as a complementary technique for enhancing the learning experience, and for stimulating the students' motivation.

6 Discussion

The experience along these years is that CnBL does not introduce any special kind of overload in the lab work from the student or from the teacher point of view. On the one hand, from the student point of view, there is no additional effort compared with any other learning technique, as the student groups have to do their practices collaboratively, and at the end participate in a tournament that takes about half an hour in our case. On the other hand, from the teacher point of view, the teacher has a natural overload the first time that the lab work is designed and implemented; as happens with any new learning technique. If the idea is to design a tournament in the same way it is presented in the paper, i.e., with a competition driven by a software platform that manages the game, then the only need is to program that software platform in advance. In our experience, the type of games, when using game theory tournaments, allow to modify such platform easily every course, reusing most of the code. Nevertheless, in our feeling, this minimal effort done by the teacher is highly rewarded by the interest of the students in doing a good job along the course. Besides, the tournament also helps to change the natural stress of an exam day into expectance to see the results of the tournament.

The use of competitions can raise some ethical issues about applying these techniques in education. A few experiences have been done up to now to consider competitive games in programming to promote learning in secondary and higher education, mainly in AI courses (Lawrence, R. 2004; Wallace & Margolis, 2007; Ebner & Holzinger, 2007; Ribeiro et al., 2009), and their results seem very positive. Nevertheless, in the work presented in this paper, the competition approach is applied taking into account some elements that must be pointed out as a stimulus for learning:

a) First, CnBL is oriented to participate in a tournament, but the learning result is independent of the student's score in the competition. The additional points obtained after the friendly tournament only improve the final mark of some groups, but do not affect the others negatively.

b) The tournament is performed among different groups, but inside a group, the students must collaborate to enhance their possibilities.

c) The competition itself is done at the end of the course, and there are no intermediate tournaments. This means that, a priori, all students' groups arrive at the tournament day with the same possibilities, avoiding lack of interest in non-successful players.

d) Finally, we remark that our experience with CnBL shows that it can be easily integrated with other learning techniques, complementing their particular advantages.

Finally, we may wonder how the approach presented here can be reused in other disciplines non-related with Computer Science. First of all, computers are nowadays completely integrated in education, so in any course we can take advantage of their support. Besides, there are some disciplines, where Game Theory has been extensively used like Artificial Intelligence, Economics, Biology, Social Sciences and Mathematics; where it is very natural to consider a methodology very similar to the one discussed here. The challenge comes from other disciplines located far away from mathematics or informatics. In this case, the idea should be to do not imitate exactly the approach discussed in this work, but to abstract the general concepts and their related benefits. This means that in those disciplines we can relax the computer-based support, and center the competition approach on the main concepts: friendly competitions among groups that should happen at the end of the course. Besides, tournaments can be done, for instance, in a Trivial Pursuit style, suggesting open problems to look for alternative solutions, or using any other type of serious game.

7 Conclusions

This paper introduces a framework for using Game Theory tournaments as a base to implement Competition-based Learning (CnBL), together with other learning techniques; to achieve a stronger motivation for the students and increase their learning performance. The paper also presents an extensive description of the learning activities performed along the past ten years on a very eclectic programming course where, in five of them, CnBL has been used to enhance the learning experience of the students.

The main contribution of the paper is the use of game theory tournaments to develop programming frameworks to support competition-based learning. The CnBL approach can exploit the facilities provided by technology and competitions to cater for diverse learning styles and individual differences. The results and mainly the feedback gained from the students suggest that the CnBL methodology discussed here motivates students to improve their work by competing against instructor-defined code and/or the code of other students in a tournament environment. The qualitative opinions provided by the students and the similarity of the surveys' results indicate that the use of friendly competitions provide a strong motivation that helps to increase the student performance.

This competition approach can be used in any programming course, and not only in artificial intelligence or game theory courses; as the idea is to create any type of programming framework, and a set of players to use it as a basis for competitive tournaments. There are several scientific disciplines, where Game Theory has been extensively used, where it is very natural to consider an approach very similar to the one discussed here. Nevertheless, in disciplines far away from mathematics and informatics teachers should consider the main concepts of the approach discussed here: use friendly competitions among students' groups (where the extra points can be considered as a reward, but not as a punishment), locate the tournament at the end of the course (to avoid lack of interest in non-successful players), and adapt the competition to the discipline style and the course contents.

Among the advantages of the competitive approach we may cite: interactivity, collaborative work inside the group, active participation, challenge versus duties, and motivation for the students to explore their own topics, on mathematics and artificial intelligence techniques, to support the challenges. But probably, the best feedback obtained is the feeling of the students along the courses where CnBL has been applied, which can be shown only partially by the surveys.

References

Adams, J. (1998) Chance-it: An object-oriented capstone project for CS-1, in Proc. 29th ACM Special Interest Group on Computer Science Education (SIGCSE) Technical Symp. Computer Science Education, 10-14.

Axelrod, R. (1984) The Evolution of Cooperation.

Barrows, H.S. & Tamblyn, R.M. (1980) Problem-based Learning: An Approach to Medical Education (Springer Series on Medical Education), New York, Springer.

Becker, K. (2001) Teaching with games: The minesweeper and asteroids experience, J. Computing Small Colleges, vol. 17, no. 2, 23-33.

Bergin, S. & Reilly, R. (2005) The influence of motivation and comfort-level on learning to program. In Proceedings of the 17th Workshop on Psychology of Programming – PPIG'05, 293–304.

Binmore K, (1994) Game Theory, Mc Graw Hill.

Boss, S., & Krauss, J. (2007). Reinventing project-based learning: Your field guide to real-world projects in the digital age. Eugene, OR: International Society for Technology in Education.

Carlile, S., Barnet, S., Sefton, A. & Uther, J. (1998) Medical problem based learning supported by intranet technology: a natural student centred approach. International Journal of Medical Informatics, 50, 1-3, 225-233.

Ebner, M. & Holzinger, A. (2007). Successful Implementation of User-Centered Game Based Learning in Higher Education: An Example from Civil Engineering. Computers & Education 49, 873-890.

Fisher, R. (1930) The Genetical Theory of Natural Selection. Clarendon Press, Oxford.

Hardin, G. (1968) The Tragedy of the Commons. Science 162, 1243-1248

Hill, J., Ray, C., Blair, J. & Carver, C. (2003) Puzzles and games: Addressing different learning styles in teaching operating systems concepts, in Proc. 34th ACMSpecial Interest Group on Computer Science Education (SIGCSE) Technical Symp. Computer Science Education, 182-186.

Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? Educational Psychology Review, 16(3), 235-266.

Hmelo-Silver, C. E. & Barrows, H. S. (2006). Goals and strategies of a problem-based learning facilitator. Interdisciplinary Journal of Problem-based Learning, 1. 21-39.

Huang, T. (2001) Strategy game programming projects, in Proc. 6th Annu. Consortium for Computing Sciences in Colleges (CCSC) Northeastern Conf. Computing Small Colleges, 205-213.

Johnson, Roger T.; Johnson, David W.; Stanne, Mary B. (1985). Effects of cooperative, competitive, and individualistic goal structures on computer-assisted instruction. Journal of Educational Psychology. Vol 77(6), Dec 1985, 668-677.

Kumar, D. (2000) Pedagogical dimensions of game playing. ACM Intelligence Magazine, 10 (1).

Lawrence, R. (2004) Teaching data structures using competitive games. IEEE Transactions on Education, Vol.: 47, 4, 459-466

Maynard Smith, J. (1982) Evolution and the Theory of Games, Cambridge University Press.

Merrill, M.D. (2007) A Task-Centered Instructional Strategy. "Journal of Research on Technology in Education, 40" (1), 33-50.

Morgenstern, O. & von Neumann, J., (1947) The Theory of Games and Economic Behavior. Princeton University Press.

Morrison, J. (2004) Where now for problem based learning? The Lancet, 363, 9403, 174.

Nash, J. (1950) "Equilibrium points in n-person games" Proceedings of the National Academy of the USA 36(1):48-49.

Novak, M. & Sigmund, K. (1993) A strategy of win stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma Game. Nature (Lond.) 364, 56-58

Pillay, H. (2002) An investigation of cognitive processes engaged in by recreational computer game players: Implications for skills of the future. Journal of Research on Technology in Education, 34 (3): 336-351.

Reese, D. (2000) "Using multiplayer games to teach interprocess communication mechanisms," in ACM Special Interest Group on Computer Science Education (SIGCSE) Bull., vol. 32, 45-47.

Slavin, R.E. (1980) Cooperative Learning. Review of Educational Research, Vol. 50, No. 2, 315-342.

Ribeiro, P., Simões, H., Ferreira, M. (2009) Teaching Artificial Intelligence and Logic Programming in a Competitive Environment. Informatics in Education, Vol. 8, 1, 85–100.

Schmidt, H. G. (1983) Problem-based learning: rationale and description. Medical Education, 17, 1, 11-16.

Wallace, S.A. & Margolis, J. (2007) Exploring the use of competetive programming: observations from the classroom. J. Comput. Small Coll., 23(2), 33–39.

	Player B stays silent (cooperate)	Player B betrays (defect)
Player A stays silent (cooperate)	Each serves 6 months	Prisoner A: 10 years Prisoner B: goes free
Player A betrays (defect)	Prisoner A: goes free Prisoner B: 10 years	Each serves 5 years

Table 1: Prisoner’s dilemma classical matrix

	Player B Cooperate (C)	**Player B Defect (D)**
Player A Cooperate (C)	**3, 3**	**0, 5**
Player A Defect (D)	**5, 0**	**1, 1**

Table 2: A classical prisoner's dilemma payoff matrix. In every cell, the left number refers to player A and the second to player B.

	Player B Cooperate (C)	Player B Defect (D)
Player A Cooperate (C)	**R, R**	**S, T**
Player A Defect (D)	**T, S**	**P, P**

Table 3: A general prisoner's dilemma form matrix

```
package agents;
import java.io.Serializable;

/**  This class defines the basic interface of a player */
public class Player implements Serializable
{
protected String sName="Basic";
protected int iId = -1;
protected int iMaxRounds = 0;

/**  This method returns the String name of the player. */
public String sGetName () {
      return sName; }

/**  This method stores the int player ID (assigned by the main tool). */
public void vSetId (int iIdAux) {
      iId = iIdAux; }

/**   This method returns the int ID of the player.      */
public int iGetId ()   {
      return iId;    }

/** This method changes the String name of the player.   */
public void vSetName (String sAux){
      sName = new String (sAux);      }

/** This method tells the player the number of succesive rounds to play against an opponent.  */
public void vMaxRounds (int iMaxAux){
      iMaxRounds = iMaxAux; }

/** This method returns the action chosen by the player. It receives the
  * present round and the number of actions to choose from. This basic player
  * always returns the first action */
public int iGetAction (int iNumRound, int iNumActions) {
      return 0; }         // The basic player always chooses the first action

/**  This method informs the player about: the winner of the game,
  * its choice, its payoff, the choice of the opponent and its playoff.
  * This method will be used by every player to update its values and
  * strategy. The method is empty and it must re-written by the students.
  */
public void vResult (int iNumRound, int iMyAction, int iMyPayoff, int iOtherAction, int
iOtherPayoff)
      {}
}
```

Table 4: Example of the Player.java file template, where a basic player is defined to be inherited and rewritten by the students.

Sentence to be valuated	1	2	3	4	5	Avg (41)
The topic selected for the tournament has been interesting:	1	1	6	10	23	**4,29**
The competition approach improved the relation with other groups:	7	6	11	11	6	**3,07**
The competition approach has been an stimulus to learn more:	0	6	6	11	18	**4,00**
The competition approach followed has been adequate:	0	1	7	11	22	**4,32**
The use of Game Theory in the lab work has been interesting:	0	1	6	19	15	**4,17**

Table 5: Data from the questionnaire fulfilled in 2005. A value of 1 means the poorest valuation, 5 means the highest. The last column displays the average over 41 answers.

Sentence to be valuated	1	2	3	4	5	Avg (44)
The topic selected for the tournament has been interesting:	0	2	10	17	15	**4,02**
The competition approach improved the relation with other groups:	11	7	13	16	7	**2,79**
The competition approach has been an stimulus to learn more:	0	1	13	21	9	**3,86**
The competition approach followed has been adequate:	0	3	11	8	22	**4,11**
The use of Game Theory in the lab work has been interesting:	0	3	10	20	11	**3,88**

Table 6: Numerical data from the 2007 questionnaire. A value of 1 means the poorest valuation, 5 means the highest. The last column displays the average over 44 answers.

Sentence to be valuated	1	2	3	4	5	Avg (40)
The topic selected for the tournament has been interesting:	0	4	8	5	23	**4,18**
The competition approach improved the relation with other groups:	4	1	22	8	5	**3,22**
The competition approach has been an stimulus to learn more:	0	4	19	1	16	**3,72**
The competition approach followed has been adequate:	0	3	5	9	23	**4,3**
The use of Game Theory in the lab work has been interesting:	0	4	12	9	15	**3,87**

Table 7: Numerical data from the 2008 questionnaire. A value of 1 means the poorest valuation, 5 means the highest. The last column displays the average over 40 answers.

Sentence to be valuated	1	2	3	4	5	Avg (125)	Var (125)
The topic selected for the tournament has been interesting:	1	7	24	32	61	**4,16**	**0,95**
The competition approach improved the relation with other groups:	22	14	46	25	18	**3,02**	**1,59**
The competition approach has been an stimulus to learn more:	0	11	38	33	43	**3,86**	**0,98**
The competition approach followed has been adequate:	0	7	23	28	67	**4,24**	**0,89**
The use of Game Theory in the lab work has been interesting:	0	8	28	48	41	**3,97**	**0,81**

Table 8: Numerical data adding all the questionnaires. A value of 1 means the poorest valuation, 5 means the highest. The last two columns display the average and the sample variance over the 125 samples.

Sentence to be valuated	2005 Avg (41)	2007 Avg (44)	2008 Avg (40)	Difference over 5 (Max-Min)	Percentage %
1st	4,29	4,02	4,18	0,27	**5,4**
2nd	3,07	2,79	3,22	0,43	**8,6**
3rd	4,00	3,86	3,72	0,28	**5,6**
4th	4,32	4,11	4,3	0,21	**4,2**
5th	4,17	3,88	3,87	0,3	**6,0**

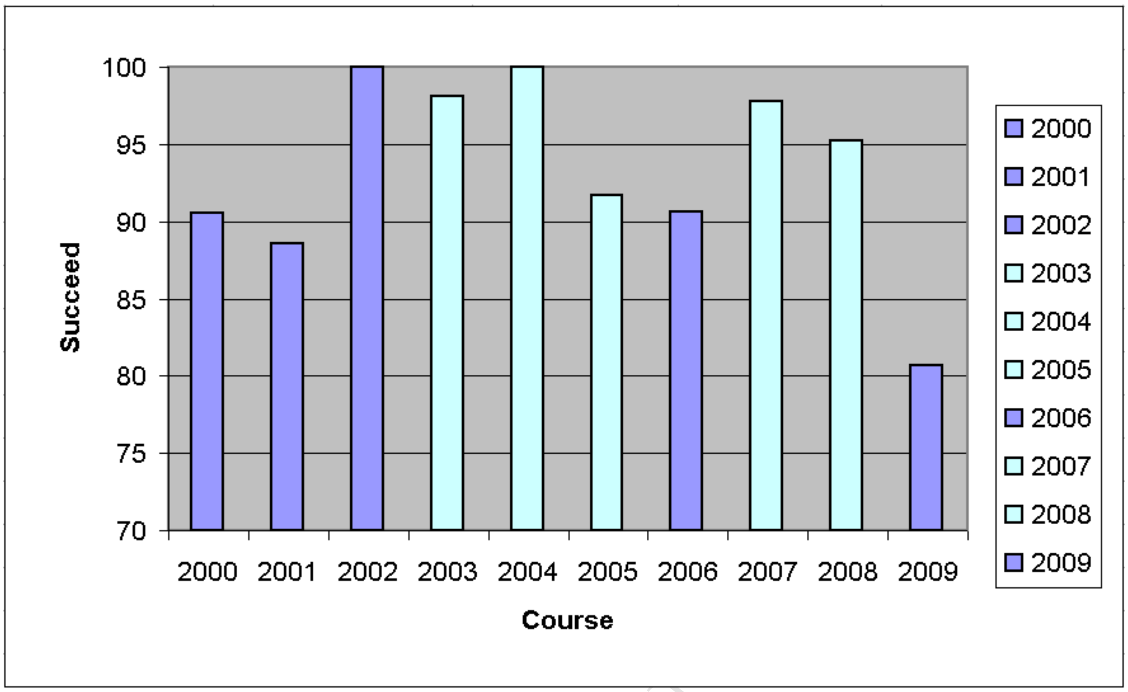
Table 9: Survey comparison, over the average values, among the courses where CnBL has been applied. The last two columns display the difference between the maximum and minimum values (over 5) and the percentage of this difference.

Course	Learning Type	Registered	% Succeed	Average Mark
2000	PjBL + PBL	53	90,57	7,88
2001	PjBL + PBL	70	88,57	8,32
2002	PjBL + PBL	46	100,00	8,27
2003	CnBL	53	98,11	8,24
2004	CnBL	42	100,00	8,73
2005	CnBL	48	91,67	8,79
2006	PjBL + PBL	75	90,67	7,36
2007	CnBL	45	97,78	8,78
2008	CnBL	42	95,24	8,47
2009	PjBL + PBL	31	80,65	7,53

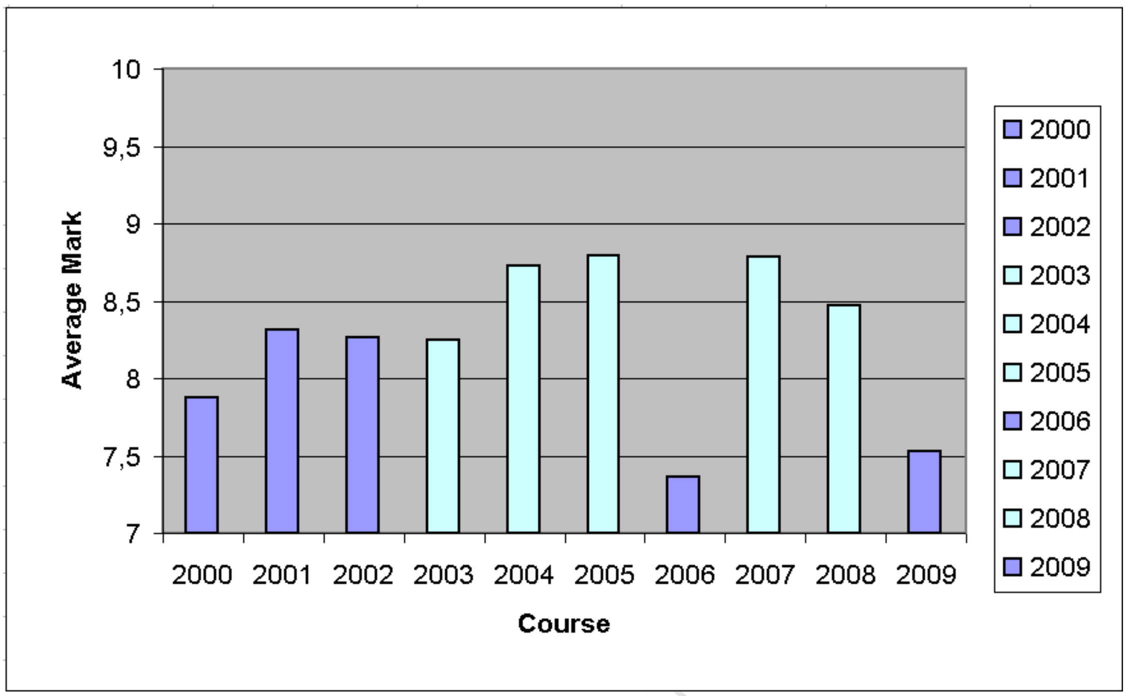
Table 10: Results obtained along the years 2000-2009. When the CnBL approach appears, it was combined with PjBL and PBL techniques.

Figure 1: Percentage of students that succeed along the courses. Courses where the CnBL approach has been performed appear in white. We point out that the graphic only shows the interval from 70% to 100%.

Figure 2: Average mark (over ten) obtained by the students along the courses. CnBL courses appear in white.



100
95
90
85
80
75
70
Succeed
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
Course


10
9,5
9
8,5
8
7,5
7
Average Mark
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009
Course
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009