PLMan: A Game-Based Learning Activity For Teaching Logic Thinking And Programming

FRANCISCO J. GALLEGO-DURÁN, CARLOS VILLAGRÁ-ARNEDO, FARAÓN LLORENS-LARGO, RAFAEL MOLINA-CARMONA

Cátedra Santander-UA de Transformación Digital.

Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, 03690 San Vicente del Raspeig (Alicante), Spain. E-mail: {fgallego, villagra, faraon, rmolina}@dccia.ua.es

This paper presents PLMan, a game-based learning activity designed to face problems observed in practical lessons about Computational Logics. The main of these problems was unmotivated students, who were showing lack of interest in learning activities. Other problems were a high percentage of students abandoning or committing plagiarism, and teachers' overload, that was leaving no time for re-designing lessons, activities and workflow. This paper describes analyses and design steps undertaken from the problematic situation to the implementation of PLMan. Experimental data confirms that this intervention reverted the problematic situation, improved learning results, raised student motivation and involvement, and left time for teachers to maintain and improve the system. Results clearly show that students have moved from literally hating activities to enjoying them and being enthusiast on participating beyond lessons.

Keywords: Serious Games; Student Motivation; Game-based Learning; Education Software

1. INTRODUCTION

The course of Computational Logics at the University of Alicante was quite similar to other first year courses in the Computer Engineering degree. Practical lessons asked students to solve two logic problems by developing two programs in Prolog programming language [1]. The goal was to introduce students to logic thinking and declarative programming. Lessons taught students basic Prolog programming and gave them hints on how to solve proposed problems.

This was the classic course scheme of Computational Logics, and it had been working for more than a decade. However, during first millenium years (2000-2005) student performance was deteriorating fast. In 2005, most students were unable to solve the problems by themselves, and required serious help. This situation led to a gigantic increase in abandon rates and plagiarism. The global situation of the subject was dramatic in terms of learning. By 2005, plagiarism peaked at 47%, whilst abandon rate stayed at 32%. Roughly 1 out of 3 students abandoned the subject, and almost 1 out of 2 staying resorted into plagirism to try to pass. These symptoms clearly were from a big learning problem that required immediate action from the teachers.

In 2004, teachers[1] conducted analyses to determine the causes and started gathering long-term data on student mood and performance. In general, students blamed Prolog: they were convinced that the language was too complex. Moreover, they perceived Prolog and Computational Logic as useless. However, most students had been able to solve problems with the same language only ten years before. That pointed to other deeper causes. Concretely, a change in students' interests appeared as the key. The generalized spread of computers, multimedia applications and games lead to a new generation of students. Students in 2005 were different from those of 1995 up to the point of having different patterns of neural connectivity [13].

The conclusion was that students had different interests. They were not motivated for learning Prolog *per se*. Solving general logic problems in a computer was not useful for them anymore. Due to this, their learning curve was turning steeper and dissolving any kind of sense of progress. They were facing uninteresting problems, with tools that they did not understand and were feeling that difficulty was too high. Most of them were simply abandoning the subject.

This paper presents the proposed solutions implemented starting from 2006 to face these learning problems, the adaptations done over time and the long term results of the overall experience. It also presents a game-based activity called PLMan [2, 3] which is the core component of proposed methodology, and may be used to reproduce this learning structure elsewhere. Section 2 sums up previous works on game-based systems considered relevant for this research. Section 3 explains the first proposed solution that was designed to improve learning results, its results, pros, cons and redesign proposals. Section 4 explains PLMan and its automated assessment system. Section 5 shows the long term results collected from 2004 to 2016 and analyzes learning impact of implemented systems. Finally, section 6 draws the conclusions of this research.

2. BACKGROUND

[1]Authors of these paper are part of the teachers. 3 of them have been teaching the subject since 1992, and one of them since 2005.

On early 2006, authors started designing new practical activities for Computational Logic. All the work started with one simple question: what are students most interested in? The intuitive answer was clear: computer games [17, 20, 25]. Literature supported the intuitive answer. Prensky [12, 14] was one of the first researchers to give relevance to this fact: computer games had been so relevant for students since 1980 that they had changed their minds. Prensky extensively treats digital game-based learning and states very relevant arguments to take into account. First, learners in the year 2000, those under the age of 36, were not the same as past learners: they were totally different in an intellectual sense. They learnt differently because they had experienced computer games for the first time in history: that was a radical new form of playing which shaped their preferences and abilities. Then, Prensky pointed out that very little research had been carried out on the intellectual differences of new gaming generations.

These differences that Prensky outlined have been getting stronger over time. More recently, McGonnigal [11] showed the importance computer games are getting and will achieve in the near future. McGonnigal defines this change as a deep revolution, which will be stronger that all previous revolutions in history together. In his view, computer games will replace a great part of reality, as millions of hours will be invested in playing them. That view expressed by McGonnigal in 2011 is clearly happening nowadays. Therefore, leaving computer games outside the classroom can potentially make lessons be part of a different reality to that preferred by most students, yielding a drop of interest in lessons.

An extensive literature review conducted by Granic et al [10] on the benefits of playing computer games shows their potential. Computer games are designed to entertain and captivate players by putting them directly in control of their alternative reality. The combination of complete autonomy, challenges, and real-time interactivity produces high intrinsic motivation that leads to high levels of concentration. This mental status produces efficient learning outcomes: players need to learn everything related to rules and game interaction to take appropriate decisions and dominate required abilities for beating the challenges.

In fact, the reasons of the success of computer games that connect them to motivational and educational properties are reported since their dawn. Bowman [8] noticed them in earlier 1982. In his own words, analyzing the game Pac-Man,

> *Pac-Man is an action system where skills and challenges are **progressively balanced**, **goals** are clear, **feedback is immediate** and unambiguous, and relevant stimuli can be differentiated from irrelevant stimuli. Together, this combination contributes to the formation of a **flow experience**.*

In this same paper, Bowman states the great achievements of Pac-Man with respect to learning,

> *It promotes **active learning** by shifting players into the participant role […] Moreover, the **immediacy** of reciprocal responses reduces the sense of distance between one's efforts and one's successes. […] Relatedly, one's efforts count for something: **status, self-determination, and sustained enjoyment**.*

These studies reinforced the idea of using computer games as a vehicle to recover lost motivation from students. However, it is important to notice that computer games are not enough by themselves. There are many examples of failures in the computer games industry. The same happens to many educational games, as Shaffer et al pointed in [7]: "*Most educational games to date have been produced in the absence of any coherent theory of learning or underlying body of research*". So, in order to be in the appropriate path to develop a solution based on computer games, it is important to know what constitutes a good game (i.e. one that will be enjoyed and played by its target audience).

In [6], Gee reports one key common factor found in many successful games,

> *Good computer and video games like System Shock 2, Deus Ex, Pikmin, Rise of Nations, Neverwinter Nights, and Xenosaga: Episode I are **learning machines**. They get themselves learned and learned well, so that they get played long and hard by a great many people*

That view reinforces the idea previously shown by Bowman [8] and also stated by Koster [15]: the fun in games comes from the fact that they can be learnt and mastered. In fact, Koster insists on a key point: "*Fun arises out of mastery. Fun arises out of comprehension. It is the act of solving puzzles what makes games fun.*" However, not everything kind of content is to be considered a game. Computer games are about experience, not about theoretical knowledge. As Gee states in [9],

> *[…] human understanding is not primarily a matter of storing general concepts in the head or applying abstract rules to experience. Rather, humans think and understand best when they can imagine (**simulate**) an **experience** in such a way that the simulation prepares them for actions they need and want to take in order to accomplish their goals*

This is exactly what good games do: good games simulate environments where players get experiences. Players love to practice and learn by experimenting and attending to feedback. Formal, theoretical knowledge is normally taught in a more direct, assertive way. That is the main reason for it not being fun

to learn: it is disconnected from imagination and experience. However, experimental knowledge develops a base for better understanding of associated theoretical knowledge, as Arena and Schwartz proof in [4].

Although there were well established ground research about game-based learning when this proposal started (2006), interest in this topic has increased exponentially in past five years (2011-2016). Numerous studies in the fields of serious games [17,18,20], game-based learning [23] and gamification [16,19,21,22,24,25] have been carried out and continue to support similar approaches.

Consequence of these studies is clear: computer games should be focused on practical knowledge and let the user experiment and learn by trial. This work starts on the ground of all these research results about computer games and their practical nature. On this basis, two game-based teaching proposals have been developed and tested showing positive results. This adds more evidence in favor of the claims previously found in the literature.

3. FIRST GAME-BASED PROPOSAL

After research on how to transform learning activities into something more interesting, it was clear that games were at the top of students' interests [10-12]. In fact, most Computational Logic students invested eight to fourteen ours per week in playing computer games. Moreover, some of them were studying computer science as a means of working in the computer games industry[2].

First step was to introduce games in the classroom to improve students' interest in term 2006/2007. As lessons were about Prolog programming, the most direct way was to change assignments into game developing. However, developing games with declarative programming was very difficult for first year students. They were not prepared to develop a game from scratch, no matter how simple the game was.

Decisions were taken to solve these issues: there will be one assignment instead of two, but split into small developing stages. Students will develop one complete game in one semester, but they will proceed step by step, with small, guided and hinted developments. These ideas concreted into a design with eight stages, and many small programming activities per stage. Students were presented at the start of each stage with a list of programming activities (as in figure 1). Each activity asked them to add a small functionality to their game (like writing a message to the screen, or asking the user what to do next). Activities were linked to their previous developments, forming an incremental structure that finished with the complete game. For each activity, they had a statement describing it and giving hints and ideas on how it should be developed. This structure was carefully designed and tested to have accurate descriptions and guide students step by step.

The main outline was as follows: first three lessons introduced students to Prolog basics and to the plot of the game. Then, each week they were presented with a new development stage. Before the start of each new stage, students had to send their completed activities as deliverables to their teachers for revision. Activities were split into mandatory and optional. All mandatory activities together formed a simple text adventure game (see figure 1, left). Optional activities improved the game by adding objects, characters and features. All mandatory activities added up to 80%, marks and optional activities to 40% marks[3].

This first proposal was our first attempt to solve motivational problems using game development, which has similar principles but is very different from the final proposal based on the PLMan game.

INSERT FIGURE 1 HERE

Fig 1. Left) Example of the minimal proposed game running. Right) Best game developed by students in 2006. Down) An example of some introductory activities on development stage 1. (Original images at the left-side, translation at the right-side)

3.1. FIRST EXPERIMENTAL RESULTS

This proposal was first implemented in the 2006/2007 term and it was operative during two terms. To measure its effect, two simple questionaries were used for students and teachers. Questionaries asked 4 yes/no questions to students and 4 quantitative ones to teachers (described in section 5). The first perceivable outcome of this implementation was an incredible turn in student motivation. The change was so dramatic that it was immediately perceived by all teachers. In 2005/2006 term, almost no student was interested in Computational Logic; in 2006/2007, 71% of students preferred Computational Logic activities to any other subject's. 25% of students showed enthusiastic about programming activities beyond lessons, and eager to know what was to come every week. Optional activities were performed by 55% of the students and 85% perceived activities as useful. Most importantly, no single student blamed Prolog for complexity: a 37% even perceived Prolog easier than other programming languages.

[2] This facts were reported by teachers out of informal interviews with students.

[3] Students were not expected to do all of the optional activities. Moreover, they were thought to let students get 100% marks even failing to get ¼ marks from the mandatory activities. This design rewarded students for working more than for being perfect, which is not to be expected from first years'.

Although these results were greatly positive, they came at an equally great cost. Designing the game and splitting it in stages and activities took approximately 150 man-hours of work (approx. 0.90 man-months). Then, each teacher was responsible for reviewing and grading every individual activity from every student. Activities were graded attending to students' code and documentation, to its clarity, structure and functionality. That added up to a mean of approx. 0.5 hours per student and week. Therefore, a teacher having 50 students required 25 hours per week to review everything. There were a total of 418 students, what made 209 man-hours per week or approx 1.25 man-months per week. That was clearly a huge drawback of the system. Moreover, feedback from teachers to students had a mean delay of 1.5 weeks. Students started new stages of development without knowing their results from the previous stage. This was also a big drawback, as feedback has a great impact on learning [4, 6, 8, 9]. These two big issues also made this system extremely costly to maintain and impossible to scale up.

Other minor issues involved problems with students' reading comprehension, which made them fail easy activities. Also, students suffered from *grading overload*: there were too many small items that added up to their final grade, and that prevented them from understanding the big picture and plan their dedication and learning strategies accordingly. Finally, the combination of these issues made some students get lost in intermediate stages and abandon (21%) or resort to plagiarism (16%).

3.2. REDESIGNING LEARNING FOR EFFICIENCY

Overall, the experience was highly positive. Teachers understood that a system like this was in the path to improving learning and satisfaction. They also noted that further development was required to overcome problems found. The two terms that the system was running established the basis for designing the next learning activities. These were the main considerations on redesign:

- **Automated grading**: the main issue with the first system was the huge overload caused to teachers. They invested all their time in reviewing and grading students' activities, and had no time for improving explanations or designing new activities. With automated grading, this huge amount of time would be free for other more effective learning tasks.
- **Immediate feedback**: as a consequence of automated grading, there was the possibility of giving students immediate feedback on their performance. This was expected to have a direct impact on learning, as students could associate causes and effects and take actions to fix their mistakes and progress on learning.
- **Adaptability and progression**: similar to mandatory and optional activities, the new system should continue offering students ways to adapt the contents to their level and interests. Enthusiastic students should have the opportunity to go beyond, while limited students should be able to progress step by step without getting lost.
- **Greater focus on logic thinking**: instead of programming, code structuring and similar abilities, one pending issue was to focus learning on logic thinking. New activities should present problems closer to logic thinking than to code structuring.

Interestingly, all these considerations are present on *good* computer games [6, 8]. Computer games liked by many people usually adapt well to different users, give immediate feedback on every action and have well designed reward systems that are summed up in their final score. This combination makes users sense that they are progressing (i.e. learning) and they are in control of the process: it is a continuous learning process based on feedback and autonomy. This analysis led redesign of the activities to be closer to game playing than to game development.

4. PLMAN: A GAME-BASED LEARNING ACTIVITY

PLMan is a Pacman-like game [2, 3, 8]. In this game students control Mr. PLMan, a character enacting Pacman, through Prolog programs called controllers. The goal is making Mr. PLman eat all the dots in a maze. Students create basic controllers with sets of simple rules like "*If a ghost is at your right, move left*". These rules are easy to understand, and can be combined to solve complex mazes requiring deep logical thinking. Initial mazes require no more than four to six simple rules. From then on, complexity of the mazes grows in manageable steps. In the end, most complex mazes require a deep sense for generalization and even some introductory concepts from Artificial Intelligence.

The game works as follows: 1) Students get presented with a new maze to solve. Figure 2 shows some maze examples. 2) Students develop controllers for Mr. PLMan in Prolog. 3) Students launch the game and test their controllers, getting the percentage of eaten dots as a performance result. 4) Students improve their controllers. 5) Repeat from step 3, until getting desired performance.

The goal is making controllers able to eat all the dots of a given maze. Mazes vary in sizes and complexity, and may include objects, enemies, puzzles and perils. Whenever students pass a maze (i.e. they complete 75% or more) they unlock the next maze. Mazes are split in 5 stages of increasing required knowledge, and grouped in difficulty levels (1-5) inside stages. They are designed to be increasingly

difficult, requiring progressively more knowledge and abilities. Starting mazes are easily solved with some simple rules. Difficulty progressively increases on new stages up to requiring abstract reasoning, generalization and planning, similar to an introductory level in Artificial Intelligence, in final stages.

INSERT FIGURE 2 HERE

Fig 2. Three examples of PLMan mazes with starting levels of difficulty. Symbol meaning: '**#**' wall, '**E**' enemy, '**O**' solid object, '**l**' gun, '**.**' dot, '**@**' Mr. PLMan.

PLMan is a turn-based game. It works similar to a classic board game, but in an electronic way. At the start of each turn, the game transfers control to the controller. Then, the game waits until the controller selects next action to be performed. When the controller returns the selected action, the game updates one complete turn (i.e. moves or changes the status of all the entities in the maze) and starts the next turn. Mr. PLMan can perform one action per turn, and is able to carry one object at a time. There are four valid actions for Mr. PLMan: move in an orthogonal direction, get an object, drop an object and use an object. In order to help taking a decision, Mr. PLMan has two sensors: a short-range visual sensor to inspect the 9 closer cells, and a long-range visual sensor, to see complete cell lists in orthogonal directions, up to the next opaque object. The game ends when all dots are eaten, when Mr. PLMan contacts a mortal entity, or when there is no more time/turns.

4.1. COMPLETE AUTOMATED SYSTEM FOR PLMAN

Along with the PLMan game, a complete automated system for assigning mazes, controlling and assessing students was developed (figure 3). 220 different mazes were introduced in this system, classified in 5 stages and 5 levels of difficulty per stage. The system asks students about the difficulty they want and assigns them available mazes. The greater the difficulty, the greater the marks rewarded. Students develop controllers at their own computers, and submit them to the system. The system runs PLMan with submitted controllers and assesses results using objective metrics like percentage of eaten dots or number of erroneous actions performed. Assessment is immediately presented to students, detailing all the metrics and executions performed. Students' marks increase with every improved controller they submit. With this incremental grading system, students know their present status and the amount of work left to achieve the grade they want.

PLMan gives immediate feedback and automated evaluation to students. Every time they launch the game they receive visual and statistical feedback on the performance of their controllers. Students start with introductory mazes and progress accordingly to their ability solving mazes. Moreover, the nature of PLMan pushes students to develop their logical thinking in order to solve presented puzzles. During their development cycle, they have to analyze results, compare with their abstract model (their mental understanding on how the controller works), understand causes of the behaviour analyzed, review their global understanding of Prolog and logic for solving the maze, and finally update the code of their controllers to reflect new knowledge acquired.

Learning model behind PLMan is based on learning by experimenting: it is an analyze-develop-test-repeat cycle. Because of that, the automated assessment system lets students submit versions of their controllers as many times as they require without penalizing. As in any computer game, the important result is being able to reach the end. It does not matter if the end is reached on the first try or after a hundred of them. Reaching the end is proof of learning by itself: students cannot reach by trial an error without learning. This has a great effect in motivation and engaging with the system, as students do not fear penalization.

INSERT FIGURE 3 HERE

Fig 3. Automated system delivers mazes to students, receives and assesses their controllers and gives them instant feedback

Finally, students may decide to stop submitting controllers whenever they wanted, and they will obtain accumulated marks. Premise is simple: submit more to increase marks, stop to get accumulated total.

5. LEARNING RESULTS AND DISCUSSION

This section presents long term results related to Computational Logics. Evidence gathered include students' general behaviour and their opinions (*S*), together with teacher effort and global perception about students (*T*). *S* items are obtained from simple yes/no questions from students, whereas *T* items are results from quantitave measures and questions from teachers. *M* items represent directly measurable data. Data starts 2 years before first redesign and continues up to 2015/2016. An average of 300 non-repeating students has entered lessons each term. Data collected is detailed as follows:

- *[M] Claims*: percentage of students that claimed about their final grade.

- *[M] Abandon*: percentage of students who abandoned the subject before the end of term.
- *[M] Plagiarism*: percentage of students that were involved in detected cases of plagiarism.
- *[M] Enthusiasm*: percentage of students that showed enthusiasm and expectation by participating in projects and aditional activities beyond lessons.
- *[M] Mean marks*: mean of the final marks obtained by all the students that finished the course.
- *[S] Prolog Easier*: percentage of students that considered Prolog as easier than other languages.
- *[S] Prefer CL*: percentage of students that preferred Computational Logic to other subjects.
- *[S] Usefulness*: percentage of students that considered the contents of the course useful.
- *[T] Revision*: hours per student and year that teachers invested in reviewing and grading deliverables submitted by students.
- *[T] Creation*: hours per student and year that teachers invested in creating new activities and contents for PLMan and the automated assessment system.
- *[T] Teaching*: hours per student and year that teachers devoted to additional teaching activities and creation/improvement of class materials.
- *[T] Mood perception*: based on the opinion of teachers, percentage of students that are motivated by the contents and methods of the course.

Results have been split in 4 charts attending to their conceptual similarities. Charts in figure 4 show a clear inverse relation between abandon/plagiarism/claim rates and student perception of Prolog being an easy language, teacher perception of student motivation and, not so strongly, the final marks obtained. Interestingly, the introduction of games as integral part of the subject in 2006 changed results dramatically. Student and teacher perception jumped from almost nothing to significant values. At the same time, abandon rates reduced more than a half, while plagiarism dropped two thirds. 2007 results seem to be a little bit worse than 2006, but still way better than 2005.

PLMan and the automated assessment system were introduced in 2008. Their introduction made plagiarism and student claims instantly vanish. At the same time, student motivation and results increased accordingly. Values reached in 2008 have approximately maintained since then on, with some variance and some curious exceptions. For instance, 2009 is a special year because a classroom contest was introduced, which gave best students additional marks. That made values like mood perception, and mean grade peak. Another curious result is the progressive decrease in teacher perception of student motivation over time. That progression seems not to be directly correlated with the other values, and might represent teachers' fatigue, probably due to lack of novelties in a long period.

Figure 5 clearly shows student reaction to the introduction of games in the classroom: their interest in the course increased from almost none to very high values. Students liked to develop games step by step, as years 2006 and 2007 show. Almost 70% of students from those two years reported Computational Logic as their preferred subject. Although an increase was predictable, these values are much higher than expected. The introduction of PLMan shows another increase, but much less steeper. However, the most important difference after introducing PLMan is the great decrease in time invested by teachers in reviewing and grading activities. That was substituted by PLMan content creation during 2008, and partly in 2009 and 2010. PLMan system was new, so content needed to be created and tested. After that, content creation has continued, but balanced with system maintenance. Revision and grading are now performed by the system, leaving free time for teachers to combine system content creation with teaching tasks.

INSERT FIGURE 4 HERE

Fig 4. Both redesigns of the course show a great beneficial impact. Claims, Abandons and Plagiarism have decayed to insignificant or normal levels. Most students changed into thinking of Prolog as an easy language, and teachers perceived a strong increase in student motivation towards the subject. Final marks have also increased significantly, according to other measures.

INSERT FIGURE 5 HERE

Fig 5. Left chart shows values related to student motivation towards Computational Logic contents. Right chart shows the teachers' distribution of invested hours.

Considering original goals, these results are highly satisfactory. Student motivation has improved beyond expectation: most of the students have been involved in the course as if it was a joyful game. Globally, students did much more work than before, and most of the work done was self-motivated. Also, their perception of usefulness has changed completely. Moreover, teachers have been completely freed of revision and grading tasks, which enabled them to do more educative-oriented tasks, like improving the system, creating new mazes, improving class materials or directly supervise students. Also, an interesting

side-effect has been produced: teaching and assessment have been decoupled, letting teachers improve their educative relationship with students without fearing to become biased for grading. A clear consequence of this is the reduction of student claims about final marks to zero. Also, all system design now collects tons of information about students' progress, constituting a highly valuable tool for analysis and improvement. Finally, plagiarism has been also reduced below 5%, not being a problem anymore.

On the other side, a great development cost had to be assumed to achieve a better and stable system. Even assuming that development cost pays off in terms of results, this initial investment represents an entry barrier that greatly depends on circumstances. This is an important point to consider before entering a redesign loop as proposed in this work.

Also, figure 5 could be anticipating a probable decline of the interest in the system. For similar reasons to the initial system, next generations of students may have different interests. This could push the whole system down. Although it does not seem probable that the system ends up with values similar to 2004 in the near future, continuous innovations are required to maintain it valid over time.

6. CONCLUSIONS

In this paper we have presented the solutions adopted to involve students into the subject of Computational Logic at the University of Alicante. Initial analyses concluded that the main problem was a change in students' interests. Students grew up playing games and they were used to immediate feedback and an accurate sense of progression. Activities lacking both become less interesting for them. Therefore, solutions presented were game-based and were custom developed for the problems analyzed.

The first solution consisted on moving assignments from solving logic problems in a computer to programming games in Prolog language. This solution yielded good results in terms of learning, but presented new challenges. It required an excessive amount of time for reviewing and grading and was very difficult to adapt to different student needs.

Redesign after the first solution produced the game PLMan and an automated assessment system. This solution transformed classroom activities into playing PLMan. Student motivation and involvement turned high and abandon rates, claims and plagiarism dropped. This latest system has been running for 8 years with similar results, confirming its validity.

Results from this particular experience are satisfactory. However, experience shows that it presents several difficulties to be applied in other fields:

- It has a high initial cost due to development and implementation.
- It requires finding a good way to transform contents into similar or equivalent game-based activities. This is a difficult task, and often implies managing non-scientific concepts like fun.
- It usually requires computer engineering professionals, with knowledge about games and game developing.

It is still pending to study the long-term validity of different game-based approaches like PLMan. Results presented here show some signs of exhaustion. Probably, most game-based activities will require continuous maintenance and innovation to keep learning results high. Future changes in students' interests may force redesign cycles. Questions on these topics remain open for future research.

In the near future, our proposed plan consists in designing an innovative in-game way to obtain data about student opinions, mood, and engagement. Getting more data for medium and long-term studies is the first priority. Then, we will proceed to integrate Machine Learning for predicting student results, estimating difficulty of mazes and pursue an optimal student-maze mapping to improve performance.

REFERENCES

1. J. Wielemaker and T. Schrijvers and M. Triska and T. Lager, Swi-prolog. *Theory and Practice of Logic Programming*, 12(1-2), 2012, pp. 67–96.
2. PLMan code repository, https://github.com/Matematicas1UA/plman, Accessed 15 October 2016.
3. F. J. Gallego-Durán, *Estimating difficulty of learning activities in design stage: A novel application of Neuroevolution,* PhD. Thesis, University of Alicante, 2015.
4. D. A. Arena and D. L. Schwartz, Experience and explanation: Using videogames to prepare students for formal instruction in statistics, *Journal of Science Education and Technology*, 23(4), 2013, pp. 538–548.
5. P. J. C. Adachi and T. Willoughby, More than just fun and games: The longitudinal relationships between strategic video games, self-reported problem solving skills, and academic grades, *Journal of Youth and Adolescence*, 42(7), 2013, pp. 1041–1052.
6. J. P. Gee, What video games have to teach us about learning and literacy, *Computer in Entertainment*, 1(1), 2003, 1-4.

7. D. Shaffer and K. Squire and R. Halverson and J. Gee, Video games and the future of learning, *Phi Delta Kappan*, 87(2), 2005, 105–111.
8. R. F. Bowman, A Pac-Man theory of motivation. Tactical implications for classroom instruction, *Educational Technology*, 22(9), 1982, 14-17.
9. J. P. Gee, Learning by design: Good video games as learning machines, *E-Learning and Digital Media*, 2(1), 2005, 5–16.
10. I. Granic and A. Lobel and R. C. M. E. Engels, The benefits of playing video games. *American Psychologist*, 69(1), 2014, 66-78.
11. J. McGonigal, *Reality is broken: Why games make us better and how they can change the world,* Penguin Group, New York, 2011.
12. M. Prensky, *Digital game-based learning,* McGraw-Hill, New York, 2001.
13. D. H. Han and S. M. Kim and S. Bae and P. F. Renshaw and J. S. Anderson, Brain connectivity and psychiatric comorbidity in adolescents with internet gaming disorder, *Addiction Biology*, 2015.
14. M. Prensky, *Don't Bother Me Mom–I'm Learning!,* Paragon House Publishers, 2006.
15. R. Koster and W. Wright, *A Theory of Fun for Game Design,* Paraglyph Press, 2004.
16. S. Azmi and N. A. Iahad and N. Ahmad, Gamification in online collaborative learning for programming courses: A literature review, *ARPN Journal of Engineering and Applied Sciences*, 10(23), 2015, 18087-18094.
17. E. A. Boyle and T. Hainey and T. M. Connolly and G. Gray and J. Earp and M. Ott. and J. Pereira, An update to the systematic literature review of empirical evidence of the impacts and outcomes of computer games and serious games, *Computers and Education*, 94, 2015, 178-192.
18. A. Calderón and M. Ruiz, A systematic literature review on serious games evaluation: An application to software project management, *Computers and Education*, 87, 2015, 396-422.
19. I. Caponetto and J. Earp and M. Ott, Gamification and education: A literature review, *Proceedings of the European Conference on Games-based Learning*, 2015.
20. T. M. Connolly and E. A. Boyle and E. MacArthur and T. Hainey and J. M. Boyle, A systematic literature review of empirical evidence on computer games and serious games, *Computers and Education*, 59(2), 2012, 661-686.
21. S. De Sousa Borges and V. H. S. Durelli and H. M. Reis and S. Isotani, A systematic mapping on gamification applied to education, *Proceedings of the ACM Symposium on Applied Computing*, 2014.
22. D. Dicheva and C. Dichev and G. Agre and G. Angelova, Gamification in education: A systematic mapping study, *Educational Technology and Society*, 18(3), 2015, 75-88.
23. M. Minović and M. Milovanović and D. Starcevic, Literature Review in Game-Based Learning, *Communications in Computer and Information Science,* 278, 2013, 146-154.
24. J. R. Morelock, Systematic literature review: An exploration of gamification in the context of engineering education, *IIE Annual Conference and Expo*, 2013.
25. G. Surendeleg and V. Murwa and H. K. Yun and Y. S. Kim, The role of gamification in education a literature review, *Contemporary Engineering Sciences*, 7(29-32), 2015, 1609-1616.

SHORT BIOGRAPHIES

Francisco José Gallego Durán obtained his B.Sc. and M.Sc. in Computer Engineering in 2003, and his PhD in Computer Science in 2015 from the University of Alicante (UA). He is professor at the UA since 2005 and his research interests are in Artificial Intelligence, Machine Learning, Games, and educative innovation. He has designed, implemented and directed the development of many games (Mad University, Screaming Racers, P84Attack, PLMan, MindRider, La Plantación…) and game engines (Wise Toad Framework, CPCtelera). Since 2008 is CTO at ByteRealms (game development trademark), and has directed the technical development of national projects like GameLearning or Time4Learning.

Carlos José Villagrá Arnedo received his B.Sc. and M.Sc in Computer Science from the Faculty of Computer Science at the Polytechnic University of Valencia in 1994, and a PhD in Computer Science at the UA in January 2016. He is currently a professor of Computer Science and Artificial Intelligence department at the UA since 2002. He was head of studies of the Multimedia Engineering degree from 2010 to 2013. His research focuses on educational potencial of Artificial Intelligence and Games.

Faraón Llorens Largo obtained his B.Sc. and M.Sc. in Computer Science in 1993, and a PhD in Computer Science by the UA in 2001. He also has a B.Sc. in Education since 1982. He has been head of the Higher Polytechnic School of Alicante (2000-2005) and Pro-Vice-chancellor of Tecnology and Educative Innovation at the UA (2005-2012). He is now head of the Santander-UA Digital Transformation Chair. He has received many education-related awards, like the *Professional Sapiens 2008* award, from the Official Association of Computer Scientists of Valencia, or the *AENUI award to*

educative quality and innovation 2013. He is currently professor at the UA and his research focuses on Artificial Intelligence, games and Gamification to improve education.

Rafael Molina Carmona received his B.Sc. and M.Sc. in Computer Science from the Polytechnic University of Valencia, Spain in 1994, and his Ph.D. in Computer Science from the UA, Spain in 2002. He is a professor at the UA and his research interests are mainly in Artificial Intelligence applied to computer-aided design, manufacture and computer graphics, game-based learning and Gamification. He has published more than 20 papers and he has also directed three Thesis in these fields. Moreover, he is now participating in a powerful research line about technology-enhanced learning and creativity, including videogames, gamification and learning analytics. He has co-authored more than 10 papers and he has co-directed one Thesis in this field.

LIST OF FIGURES

Figure 1

Figure 2

Figure 3

Figure 4

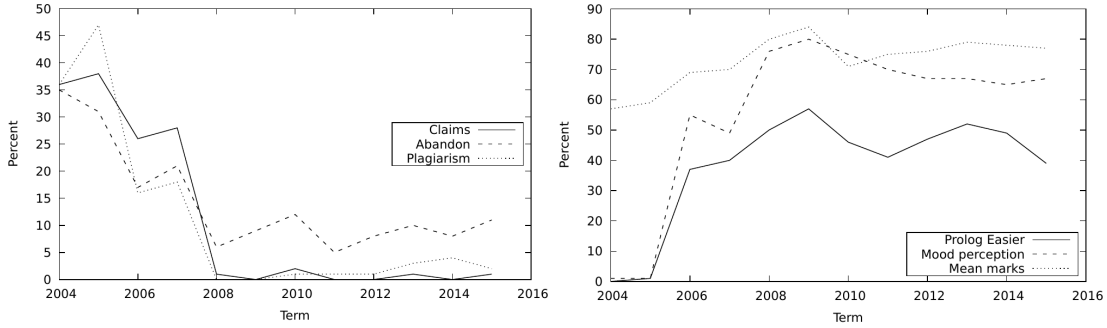

Figure 5