Teaching Aspects of Constraint Satisfaction Algorithms Via a Game

Ioannis Hatzilygeroudis, Foteini Grivokostopoulou, Isidoros Perikos
School of Engineering
Department of Computer Engineering & Informatics
University of Patras
6500 Patras, Hellas (Greece)
{grivokwst,perikos,ihatz@ceid.upatras.gr}

Abstract

In an Artificial Intelligence course, a basic concept is Constraint Satisfaction (CS), which is acknowledged as a hard domain for teachers to teach and student to understand. In this paper, we present a game based learning approach to assist students in learning CS algorithms, such as arc consistency and search algorithms, for problem solving in an easy, interactive and motivating way. Preliminary valuation has showed promising results.

Introduction

Computer games have been used successfully in both introductory computer science courses (Bayliss 2007) and general artificial intelligence classes (Wong, Zink, and Koenig 2010). Indeed, through games students are given motives to increase their interest and teachers implement the learning by doing or learning by participating principle. So, students have a framework for better grasping or understanding AI concepts.

There are a number of efforts for using games in teaching AI. In (Markov et al 2006) the N-Puzzle game is used for teaching state-space search and machine learning. In (DeNero and Klein 2010) the Pac-Man is used for teaching AI concepts such as, state-space search, adversarial search, Markov decision process, reinforcement-learning and probabilistic tracking. Furthermore, the Rook Jumping Maze has been used for teaching uninformed search, stochastic local search and machine learning (Neller 2011). A common characteristic of the above efforts is that games are mainly involved in AI programming projects. A project is designed based on a game that asks for implementation of algorithms concerning state search, machine learning, etc.

In this paper, we present use of a game for teaching basic concepts of constraint satisfaction (CS) problem solving. We want to use the game as a means of students creating a right cognitive model of the corresponding concepts or processes. For this purpose, we implemented the map-coloring problem (Fritsch and Fritsch 1998) as a game. The game offers an environment for learning concepts related to CS, like the arc consistency algorithms (i.e. AC-3 algorithm) and their combination with search algorithms (i.e. depth-first with backtracking). By playing the game, a student comes across concepts of CS and unknowingly emulates steps of the algorithms.

"Coloring Map" Game

We designed and implemented a computer game based on the map-coloring problem so that AC-3 and its combination with depth-first with backtracking algorithm can be simulated. The game is called "Coloring Map" and a snapshot of it is illustrated in Figure 1.

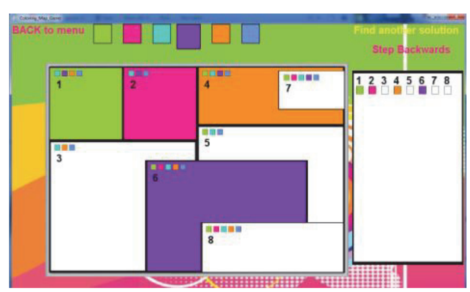

Figure 1. A snapshot of the Map Coloring game

The main goal of the game is to assist students to understand CSP solving. A student's challenge is to paint the generated maps using a set of available colors so that the adjacent regions of the map do not have the same color. The complexity of the generated maps (number of distinct areas) and the available colors set are parameterized. The "coloring map" game leaves the student to determine the complexity of the problem to be solved. This corresponds to specifying the number of discrete areas of the map and the number of colors to be used. The student can either decide on how many colors he/she thinks are adequate for solving the problem or can choose one of the three available difficulty levels. Experimenting with different numbers of colors the student becomes familiar with the constraints of the problem and can develop a sense of the relation between the map complexity and the minimum number of colors required. Also, the user can choose from the set of all available colors for each area or the system will filter the colors that cannot be used (i.e. propagate the constraints, in other words make an AC-3 step). In this way, the student gets a useful representation of how the constraints are propagated, which is an important and fundamental characteristic of solving CS problems (CSPs). Furthermore, the student can step back to previous states of the solution path, after having reached a dead end, a process known as backtracking. So, by using the game, the student can see backtracking in practice. Finally, the student can solve the problem as many times as he/she wants, using different color combinations, thus producing a set of different solution paths.

Evaluation

15 undergraduate students participated in the evaluation. All students were taught about CS problem solving and faced CSPs using the arc consistency algorithm AC-3 or depth-first with backtracking search during the course lectures. The course had three scheduled tutorials. Initially, the students had to take a short pre-test (without using the game), which included ten questions. In Table1, the domain of questions and results of the pre-test are presented.

	Pre Test Results	
Questions	Domain	(%) correctly
1 2	Representation of CSP	40
3 7	AC 3 Algorithm	32
8 10	AC 3 & Search Algorithms	33,3

Table 1. Pre Test Results

The results showed that 40% of the students created correctly the CS representation of the problem. Moreover, approximately 32% specified correctly the steps of the AC-3 algorithm and only 33,3% of them were able to correctly combine the AC-3 with a search algorithm.

After having finished the pre-test, the game became available to the students. A playing session covering instructions on how to use the game was given to the students along with a set of different (i.e. five) map-coloring problems to try. Students had one week to play with the game. After that, the students took a post-test including ten questions. The results gathered from the students' answers are presented in Table 2.

	Post Test Results	
Questions	Domain	(%) correctly
1 2	Representation of CSP	60
3 7	AC 3 Algorithm	56
8 10	AC 3 & Search Algorithms	51

Table 2. Final Test Results.

The results are quite encouraging and show that the students were able to answer correctly more questions than before had played with the game. More specifically, 60% of the students were able to create correctly the CSP representation of the given problem vs 40% in the pre-test.

Conclusion

Contemporary education may not be based exclusively on class lectures, given the current capabilities of game technology. Educational games are suitable for a more engaging, motivating and entertaining learning.

In this paper, we present a game developed to help students learn CSP solving. The game was developed to offer to the students an entertaining interactive and motivating way to experience with and learn about CSP solving. The evaluation results are very encouraging. They show that the game helped students get a better insight of the representation of CSPs and also of the implementation of the AC-3 and the search algorithms.

Acknowledgement

This work was supported by the Project "Research in Informatics" (No C355, Research Committee of the University of Patras, Greece).

References

Bayliss, J. D. 2007. *The effects of games in CS1 3.* Journal of Game Development 2.

DeNero, J., and Klein, D. 2010. Teaching introductory artificial intelligence with pacman. In Proc. of the 2nd EAAI Symposium, 1885 1889.

Fritsch, R. and Fritsch, G. 1998. *The Four Color Theorem, History, Topological Foundations and Idea of Proof*, Springer Verlag, New York.

Markov, Z., Russell, I., Neller, T., Zlatareva, N. 2006. Pedagogical Possibilities for the N Puzzle Problem. In Proc. of the 36th ASEE/IEEE Frontiers in Education, S2F 1 S2F 6.

Neller, T. W. 2011. Rook Jumping Maze Generation for AI Education. In Proc. of the 24th FLAIRS Conference, 382 387.

Wong, D.; Zink, R.; and Koenig, S. 2010. Teaching artificial intelligence and robotics via games. In Proc. of the 2nd EAAI Symposium, 1917 1918.