Prolog Project 2016-2017

Elisa Fromont

September 20, 2016

1 Subject

Your aim is to implement a 2-player game in **Prolog** (possibly using constraint programming, but this is not mandatory). The project will be done by teams of 3 to 4 persons. The defenses will be held in January 2017. Each team can chose **any 2-player game** (there are some suggestions in the folowing section).You need to send me by mail the **project members** before the **14/10/2016** and the game you have chosen before the **4/11/2016**. If you do not send me any mail (at the given deadlines), you will be assigned a team and a game arbitrarily (without any possible changes).

The project consists in programming the game in Prolog (and/or CLP) to make it possible for a player to play against the computer. To win the game (and to allow a real interaction), the computer should need to use good search strategies and in particular good heuristics. There should be at least one heuristic encoded by each member of the team (e.g. 4 different heuristics to win for 4 members). The use of the heuristic should be a parameter of the program and the details of each heuristic will be asked during the project defense. Implementing a GUI is not mandatory (but could be nicer) as long as it is possible to write a move in the command line. The GUI does not need to be implemented in PROLOG if you can find good ways to interface your chosen language with Prolog ex: `http://www.swi-prolog.org/packages/jpl/java_api/`). Note that you need most of the knwoledge of your AI course (prolog programming, CLP, search algorithms, ...) to do the project so you may not be able to start it before November.

Game Programming You can find more information about Game-Playing Programs in Chapter 21 of "The Art of Prolog" from Leon Sterling and Ehud Shapiro (MIT Press 1994) (`http://perso.univ-st-etienne.fr/frel9915/Prolog/chap21_artOprolog.pdf`).

2 Plagiarism

As you will soon find out, many 2-player games have already been implemented in Prolog and can be found on Internet. For example Mastermind, Awele, (ultimate) Tic tac toe, (Chinese) Checkers, Othello, Abalone, Battleship, Connect four (or more), SIAM, Nim games or even Rasende roboter (`http://www.trictrac.net/jeu-de-societe/rasende-roboter/infos`) are almost all available on github. Some games are more difficult to find such as Blokus, Genial, Quarto or Corridor. You can find ideas (and rules) on this page `http://boardgamegeek.com/geeklist/59830/abstract-thinking-your-favorite-two-player-abstrac`.

If you chose a game for which the code is available on the Internet, you will have to cite your sources and your project will have a maximum grade of 15/20. Every project needs to have at least 40% of original code. Projects which have 100% of original code will be graded up to 20/20.

Projects which use a code available on the Internet or in a book without citing their sources will be graded with a 0/20 if discovered.

3 Evaluation

The defense will be a 30 minutes demo where I should be able to play against your program.

Before the defense, each team should put 2 days before the defense (deadline at midnight) on claroline (submission after the deadline will be graded with a 0/20):

1. A commented source code;
2. A report (max 4 pages) which describes the programmed game, its rules, the description of the different heuristics, the sources (links) of all the non original material, a report on the efficiency of the algorithms and a set of tests for each heuristic implemented.

Here are the points that will decrease your grade (not uniformly):

1. We have not heard about you or your team before the project defense;
2. The project has not been sent before the defense;
3. The code is of poor Prolog quality (you programmed in an "imperative" way);
4. The code is not commented (even it is not your own code);
5. The project has not been tested or the tests are not available in the report;
6. The project does not work (it is impossible to play);
7. The project use some code from the Internet (and worse, more than 60% or even worse, the sources are not cited);
8. There is less heuristics implemented than the number of members in the team;
9. Some students in the team have not done anything;
10. The computer takes more than X seconds to play a move (the higher the X, the worse the penalty);
11. The team has invented a trivial game;
12. The computer plays stupidly;
13. The demo was not convincing (e.g. not prepared).

If none of the above points apply to your team, your grade will be 20/20.