Fast and Furious Game Playing: Monte Carlo Drift Planning report

Prateek Bhatnagar, Baptiste Bignon,
Mikaïl Demirdelen, Gabriel Prevosto,
Dan Seeruttun--Marie, Benoît Viguier

Supervisors: Nikolaos Parlavantzas, Christian Raymond

12/19/2014

Contents

1 Introduction

The project is called *Fast & Furious Game Playing, Monte Carlo Drift*. Its purpose is to create an Artificial Intelligence able to compete against humans using the *Monte Carlo Tree Search* algorithm.

We have chosen the game Arimaa because it is a two-players strategy board game not solved[1].

A human plays a game by thinking of all possible moves as per one's imagination and then opts for the best amongst them. Before taking ones turn, a player can visualize ones options and predict how an opponent will counteract them. The algorithm will do the same by building a search tree containing the different posibilities. The Minimax algorithm does it by exploring all possibilities, which is heavy. The MCTS algorithm is lighter and converges to the Minimax algorithm, therefore it has been chosen for this project.

By exploring the numerous random possibilities at any one time, our program will be able to take decisions in order to win the game. The algorithm will be parallelized in order to exploit it in a multi-core machine, allowing it to go further into the search tree, thus improving its efficiency.

The planning methods we used will be explained. Furthermore, we will develop the context of our environment namely we will precise the main dates on our calendar and the ressources we will have. Then, we will plan the different tasks we will realise along the project. With those tasks and the dependencies between them, we will create a Gantt diagram in order to graphically represent our planning. In order to reduce brakes on the project, we will analyse the different risks about it.

[1] A game solved is a game where good algorithms are able to find the perfect move in each situation to win, or to draw. For instance, *Tic Tac Toe* or *Draughts* are solved games.

2 Planning Methods

2.1 Presentation Methods

We are using for our project the Agile method and the Poker planification method.

The Agile method is a time bounded and iterative software development methodology for building and testing the software incrementally from the start of the project, instead of trying to develop and test it all at once near the end.

To initiate the development of the project, each group member sits together and prepares a list of features they would like to see in their software.These things are called as user stories and they become the*To Do* list of the project.Thereby using the Agile estimation techniques (such as Poker method),the stories are sized relatively to each other, by coming up with a guess as to how long will each user story take to complete.

Before development, the tasks in the list that are to be done are prioritized so that the most important stuff gets the highest priority and the least important one can be left for the last. Then the development of the project starts with the initiation from top to bottom by building, iterating and getting feedback from the end user or the development team as the development proceeds.

2.2 Environment

2.2.1 Human Resources

Our project group is composed of six students, half of which will be abroad next semester. That implies that most of the development phase will be carried out by a team of three people, with the help of the supervisors. We also hope to get some advice from Mr Garcia, a researcher from the INSA who knows a lot about Artificial Intelligence.

2.2.2 Other resources

The algorithm is due to work on a set of clusters of multi-core machines. Then, we will test is on the computer of our INSA Computer Science department. *Grid'5000* is as well a set of clusters of multi-core machines, and we hope to get the time and the authorization required to use it to test our algorithm.

2.2.3 Exterior knowledge

The Artificial Intelligence (AI) will implement the *Monte Carlo Tree Search* ($MCTS$), an algorithm that has been used in the past for board games. The AI will be parallelized using the *Root parallelization* strategy, one of the main strategies used with the MCTS algorithm.

2.2.4 Technologies used

A few technologies will be used for this project:

- The project will be coded in C++ language
- The application used as user interface uses SFML 1.6
- MPI, OpenMP and OpenAcc will be used for parallelization

2.2.5 Important dates

For this project, a number of set deadlines exist:

- A conception report is due by the 12^{th} of February
- A HTML documentation is due by the 25^{th} of March
- The final report is due by the 26^{th} of May
- A demonstration of the AI will take place on the 26^{th} of May
- The application is to be finished and sent by the $2^{th}8$ of May
- A presentation of the project will take place on the 28^{th} of May

In order to give us more time to work on the project, some time has been freed on our schedule, from May the 18^{th} to May the 28^{th}. On the other hand, the exams that take place on the weeks of January the 12^{th} and May the 11^{th} have been considered as time off the project. Outside of these time periods, the workload for each member of the team has been set to 4 hours a week, in order to increase our flexibility.

3 Task Planning

3.1 Tasks list for Gant diagram

3.1.1 Tasks gathered in modules

To understand better how the tasks are organized, tasks have been classified in modules, representing the main aspects of our project. Each task will be checked, and tested just before the end of the test, according to the Agile method.

- The first module is the GUI application. It contains a Graphical Interface for our game, but as well the rules to play, and a recorder of rules. This application will make it possible for an human to play against another human.
- The second module is the MCTS algorithm. It is the heart of our project. This algorithm will be used to compute a set of all moves possible ordered in a tree. First, it will be implemented on *Tic-Tac-Toe*, then on *Connect4* and finally on *Arimaa*.

  With this module, it will be possible to play against the AI. There will be as well some improvements to do as the use of Boost library, a better memory management, and making statistics.
- The module Converter is a converter of data to make communicate the MCTS part and the GUI part.
- The Parallelization module represents the management of our environment. Our algorithm will be tested using different computer powers, with CPU and GPU parallelization. It will be tested on our computers, on the computers of our INSA computer science department. Finally, if the time permits us, we will test it on the researchers' network *Grid'5000*.

  This module will increase the performances of our algorithm, giving it calculus power.
- The last module is the Documentation. It will take a lot of our time, writing 2 reports, preparing 2 oral presentations, conceiving an HTML page and a CD of the application.

All these tasks form part of the Gantt diagram.

3.1.2 Gantt diagram

TODO baptiste

3.2 Risks analysis

There will always be problems or delays that come up. The best way to deal with them is to think about what could go wrong in the project rather than waiting to get into deep

water. The 5 kind of risks are the following:

1. Technical
2. Ressources
3. Organisation
4. Payments
5. Suppliers/Purchases

Our main risk factor is the point number 1 : technical. We are not dependent to anything such as purchases, ressources or organisation as we are a small group of workers.

To get into further details, here is a list of what might possibly become cumbersome:

- getting used to the technology we will use (CAF, OpenMPI, OpenACC, OpenMP, Boost Library...)
- booking the use of *Grid'5000*, we do not know how long is the waiting list in order to be able to use it.
- Interoperability problems: most of us work on *Windows* operating systems. However *Grid'5000* runs on Linux, for this reason we will test our algorithm on a smaller scale. The computer science departement at INSA will be our first testing facility in order to make our cluster implemtation works.

Some other problems might come up later and we will try to deal with them as soon as possible to get the least delay as we possibly could.

4 Conclusion

The main focus area of this report is planning. Various planning methods have been finalised such as Agile development methodology and Planning Poker method. Using Agile Development Methodology, the incremental development of the project will take place. Planning Poker method which is the estimation technique of Agile Methodology is used to estimate the time requirement for the development of the project. The dates have been estimated with respect to the development of various models of our project. We will work over hours a week for our project. We will try to foresee most of the risks, and if it is not the case, they would be added in our planning with Microsoft Project.