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.

This algorithm will be parallelized in order to optimize it in a set of multi-core machines, 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 the 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 Development Methodology

2.1 Agile Methodology

The development methodology used for this project is Agile Development Methodology with Planning Poker estimation technique.

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 Planning Poker

Planning poker is a consensus-based agile estimation technique. Planning Poker is very simple and effective. It is being utilised by the agile teams around the world to estimate their product backlogs. It can be used with story points, ideal days, working hours or any other estimating unit.

In planning poker, each individual in the team is asked to put an estimation about the tasks related to the project such as estimation about the consumption of time, or estimation about the resources required. Then keeping in mind the consensus, the most voted value or the value nearest to the most voted value about the task is chosen. The singular value are ignored.

Planning Poker reminds everybody that the estimate includes all of the work that needs to be accomplished in order for the story to be considered done. It includes development, testing, documentation, and anything else required for completion.

3 Environment and risks analysis

3.1 Environment

3.1.1 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. The planning has been done assuming that we would not recieve help from people outside the project, but we plan to ask Mr Garcia, a researcher from INSA, for advice. Since he knows a lot about Artificial Intelligence, his help would help speed up the project.

3.1.2 Other resources

The algorithm is due to exploit parallelization in order to give more accurate results. We will first test it on the machines from INSA's Computer Science department; then, if we have the time and the authorization required, we will run it on *Grid'5000*, a set of clusters of multi-core machines.

3.1.3 Exterior knowledge

The Artificial Intelligence (*AI*) will implement the *Monte Carlo Tree Search* (*MCTS*) 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.

3.1.4 Technologies used

A few technologies will be used for this project:

- The project will be coded using the C++ language
- The application providing a user interface uses SFML 1.6
- Either MPI, OpenMP and OpenACC or an implemntation of the actor model (such as *CAF*) will be used for parallelization

3.2 Risks analysis

Problems or delays are inevitable. The best way to deal with them is to predict what could go wrong in the project rather than waiting to get into deep water. The 5 categories below are those in which risk is most likely to occur :

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

The main risk factor is the first item : technical. We are not dependent on anything such as purchases, ressources or organisation as we are a small group of workers.

A list of what might possibly become cumbersome may now be established :

- getting used to the technology we will use (CAF, OpenMPI, OpenACC, OpenMP, Boost Library...).
- unexpected bugs in the program.
- booking the use of *Grid'5000*.
- 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. Our cluster implementation will first be put to the test at the Computer Science department at INSA.

In order to avoid theses difficulties, we made sure to dedicate enough time to study the technology used and to fix the bugs. We will also make sure to book *Grid'5000* early. Were it not be avialable, our tests would be done on a set of clusters from INSA's Computer Science department. While the computing power is not comparable to *Grid'5000*, we still expect to get reliable results. A monthly test of our implementations in the Linux rooms of the department will guarantee us that our application does not have any interoperability problems.

Some other problems might come up later and we will try to deal with them as soon as possible to avoid unnecessary delays.

4 Task Planning

4.1 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 28^{th} 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.

4.2 Tasks list for Gant diagram

4.2.1 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 after, 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.

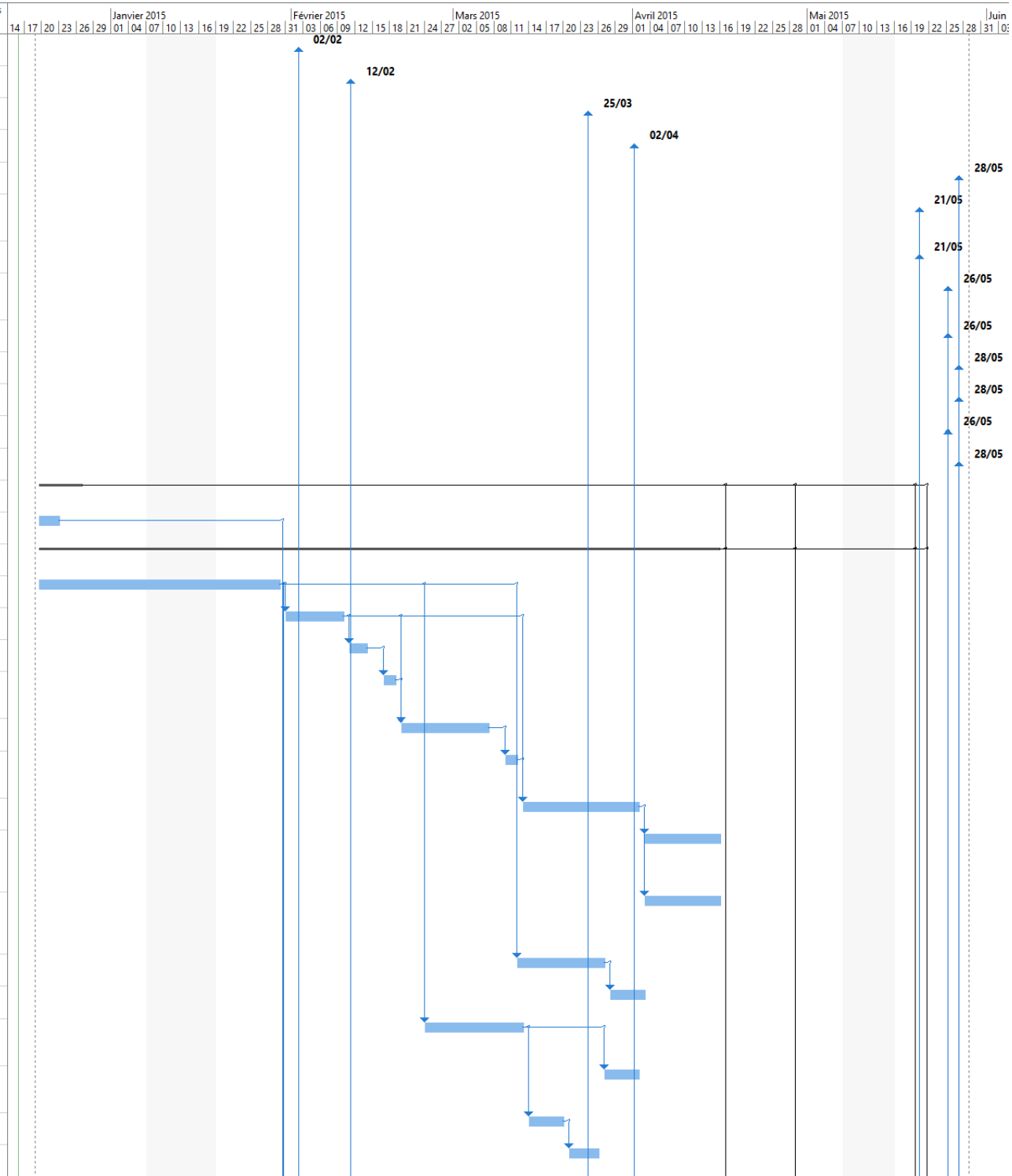
4.2.2 Gantt diagram

The Gantt diagram is situated at the end of the report.

5 Conclusion

The main focus area of this report is planning. Various planning methods have been decided 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. A serious attempt is made analyse the risks in order to take preventive measures to avoid various risks and problems.

N°	Mode Tâche	Nom de la tâche	Durée	Début	Fin	Prédécesseurs	Noms ressources
1		Conception report first draft	0 jour	Lun 02/02/15	Lun 02/02/15	47	
2		Conception report last draft	0 jour	Jeu 12/02/15	Jeu 12/02/15	48	
3		HTML page	0 jour	Mer 25/03/15	Mer 25/03/15	49	
4		HTML page final version	0 jour	Jeu 02/04/15	Jeu 02/04/15	50	
5		Documentation	0 jour	Jeu 28/05/15	Jeu 28/05/15	51	
6		Planning assessment report first draft	0 jour	Jeu 21/05/15	Jeu 21/05/15	52	
7		Final report first draft	0 jour	Jeu 21/05/15	Jeu 21/05/15	53	
8		Planning assessment report last draft	0 jour	Mar 26/05/15	Mar 26/05/15	54	
9		Final report last draft	0 jour	Mar 26/05/15	Mar 26/05/15	55	
10		Project presentatio	0 jour	Jeu 28/05/15	Jeu 28/05/15	56	
11		Deadline for the application	0 jour	Jeu 28/05/15	Jeu 28/05/15	57	
12		Demonstration	0 jour	Mar 26/05/15	Mar 26/05/15	58	
13		Showroom	0 jour	Jeu 28/05/15	Jeu 28/05/15	58	
14		**GUI Application**	**9 jours?**	**Ven 19/12/14**	**Ven 26/12/14**		
17		Test CAF on Windows	5 jours?	Ven 19/12/14	Lun 22/12/14		Groupe
18		**MCTS algorithm**	**79 jours?**	**Ven 19/12/14**	**Mer 15/04/15**		
19		MCTS general algorithm	25 jours?	Ven 19/12/14	Jeu 29/01/15		Groupe
20		Test of MCTS algorithm	7 jours?	Ven 30/01/15	Lun 09/02/15	19	Groupe
21		MCTS applied to TicTacToe	4 jours?	Mar 10/02/15	Ven 13/02/15	20	Groupe
22		Test of MCTS algorithm on TicTacToe	3 jours?	Lun 16/02/15	Mer 18/02/15	21	Groupe
23		MCTS applied to Connect4	12 jours?	Jeu 19/02/15	Ven 06/03/15	20;22	Groupe
24		Test of MCTS algorithm on Connect4	3 jours?	Lun 09/03/15	Mer 11/03/15	23	Groupe
25		MCTS applied to Arimaa	15 jours?	Jeu 12/03/15	Mer 01/04/15	20;24	Groupe
26		Test of MCTS algorithm on Arimaa versus humans	10 jours?	Jeu 02/04/15	Mer 15/04/15	25	Groupe
27		Test of MCTS algorithm on Arimaa versus other AI	10 jours?	Jeu 02/04/15	Mer 15/04/15	25	Groupe
28		Implement MCTS with Boost	12 jours?	Mer 11/03/15	Jeu 26/03/15	19	Groupe
29		Test of MTCS boost	5 jours?	Ven 27/03/15	Jeu 02/04/15	28	Groupe
30		Implement MCTS with memory	14 jours?	Lun 23/02/15	Jeu 12/03/15	19	Groupe
31		Test of MCTS with memory management	5 jours?	Jeu 26/03/15	Mer 01/04/15	30	Groupe
32		MCTS time test (statistics)	5 jours?	Ven 13/03/15	Jeu 19/03/15	30	Groupe
33		Test of MCTS tim	4 jours?	Ven 20/03/15	Mer 25/03/15	32	Groupe



34		**Converter**	**9 jours?**	**Ven 19/12/14**	**Ven 26/12/14**		
35		Implement the converter	6 jours?	Ven 19/12/14	Mar 23/12/14		Groupe
36		Test the converte	3 jours?	Mer 24/12/14	Ven 26/12/14	35	Groupe
37		**parallelization**	**52 jours?**	**Ven 30/01/15**	**Lun 13/04/15**	**17;19**	
38		Implement CPU parallelization	10 jours?	Ven 30/01/15	Jeu 12/02/15		Groupe
39		Test CPU parallelization	4 jours?	Ven 13/02/15	Mer 18/02/15	38	Groupe
40		Implement cluster parallelization	10 jours?	Ven 30/01/15	Jeu 12/02/15		Groupe
41		Test cluster parallelization	4 jours?	Mar 17/02/15	Ven 20/02/15	40	Groupe
42		Implement GPU parallelization	10 jours?	Lun 02/02/15	Lun 16/02/15		Groupe
43		Test GPU parallelization	4 jours?	Jeu 19/02/15	Mar 24/02/15	42	Groupe
44		Tests in depart	10 jours?	Mer 25/02/15	Mar 10/03/15	39;41;43	Groupe
45		Tests on Grid500(	7 jours?	Ven 03/04/15	Lun 13/04/15	44	Groupe
46		**Reports and presentations**	**113 jours?**	**Ven 19/12/14**	**Ven 22/05/15**		
47		Conception report first draft	20 jours?	Ven 19/12/14	Jeu 22/01/15		Groupe
48		Conception report last draft	6 jours?	Ven 23/01/15	Ven 30/01/15	47	Groupe
49		HTML page	14 jours?	Ven 19/12/14	Ven 02/01/15		Groupe
50		HTML page final version	5 jours?	Lun 05/01/15	Mer 21/01/15	49	Groupe
51		Documentation	10 jours?	Mar 19/05/15	Jeu 21/05/15	14;18	Groupe
52		Planning assessment report first draft	20 jours?	Jeu 16/04/15	Mar 19/05/15	14;18;34;37	Groupe
53		Final report first draft	20 jours?	Jeu 16/04/15	Mar 19/05/15	14;18;34;37	Groupe
54		Planning assessment report last draft	5 jours?	Mar 19/05/15	Mer 20/05/15	52	Groupe
55		Final report last draft	10 jours?	Mar 19/05/15	Jeu 21/05/15	53	Groupe
56		Project presentation	15 jours?	Mar 28/04/15	Jeu 21/05/15	14;18;34;37	Groupe
57		Creation of the CD of the application	6 jours?	Jeu 21/05/15	Ven 22/05/15	14;18;34;37	Groupe
58		Demonstration	8 jours?	Jeu 16/04/15	Lun 27/04/15	18;37	Groupe

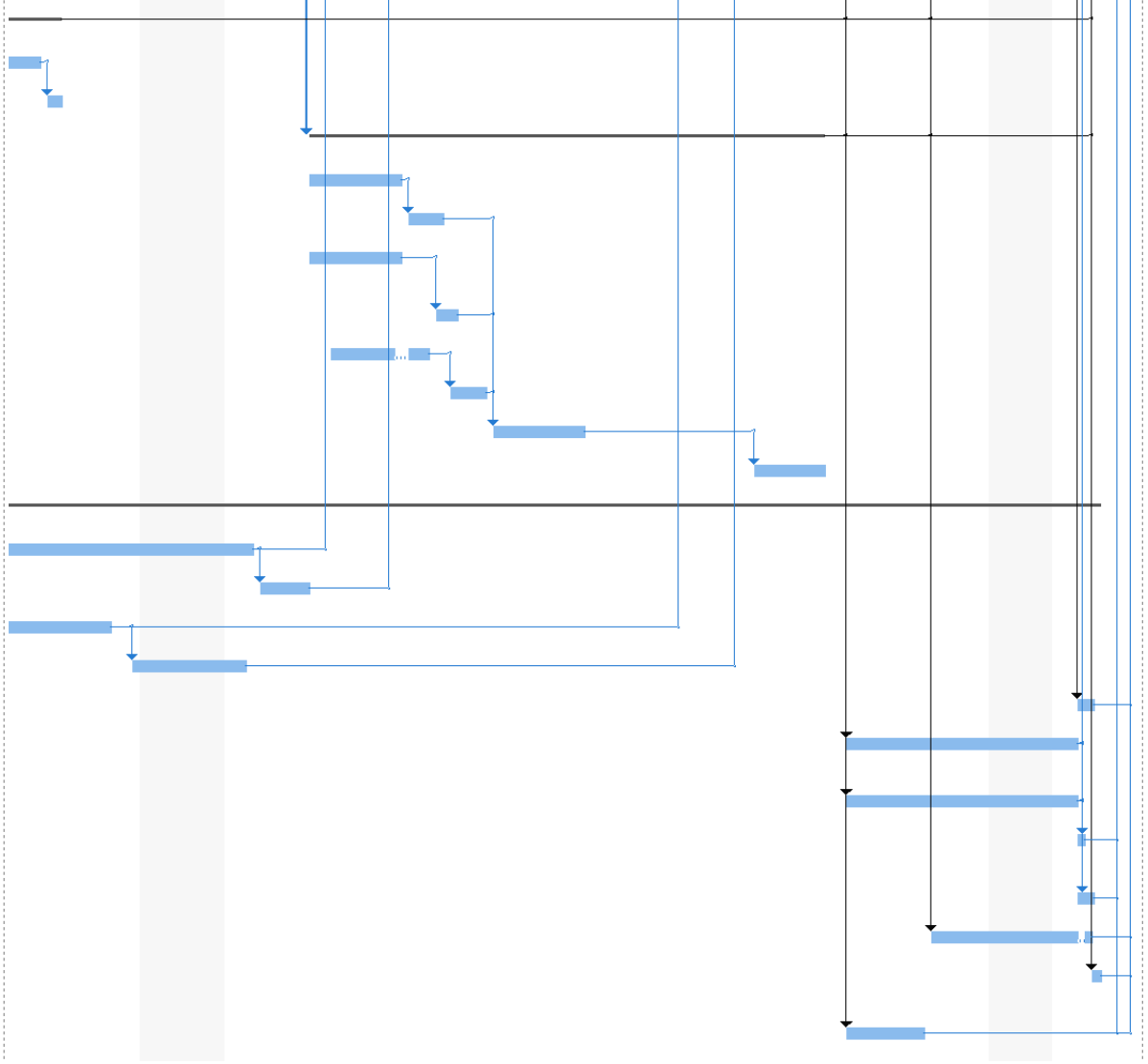


Figure 1 : Gantt Diagram