



ANALYSE ET RÉOLUTION DE CASSE-TÊTE : FLOOD-IT

Réalisé par :
BELLEL Oussama
BOURGIN Jérémy
HERRI Abdallah

Sous la direction de :
MASSEPORT Samuel
VOLTE Gabriel

Pour l'obtention du Master1 AIGLE
Année universitaire 2018/2019

Table des matières

1	Introduction	3
1.1	Généralités	3
1.2	Flood-It	3
1.3	Objectifs du projet	4
2	Planning	5
2.1	Planning final	8
3	Déroulement du projet	9
3.1	Données quantitatives	9
3.2	Données qualitatives	9
4	Conclusion	10

Table des figures

1.1	Coloration d'une grille de 3x3	4
2.1	Planning prévisionnel - Diagramme de Pert	6
2.2	Planning prévisionnel - Diagramme de Gantt	7

Chapitre 1

Introduction

1.1 Généralités

Dans le cadre du projet de TER en première année de Master AIGLE, nous devions réaliser un programme permettant de résoudre un casse-tête. Les encadrants de ce projet nous ont laissé le libre choix du casse-tête.

Après avoir essayé différents jeux, nous avons choisi le jeu Flood-It. Le choix de ce jeu s'explique par le fait qu'il permet d'avoir un nombre important d'approches différentes pour le résoudre. Ainsi, avec plusieurs méthodes de résolution, le but serait de pouvoir en réutiliser plusieurs sur d'autres casse-têtes.

Pour cela, nous avons joué plusieurs parties et essayé différentes techniques pour finir le jeu le plus rapidement que possible. Ainsi, avant même de commencer à implémenter le jeu, on avait quelques idées d'algorithmes que l'on pourrait utiliser pour ce jeu.

1.2 Flood-It

Flood-it est un jeu combinatoire : il existe plusieurs combinaisons possible pour résoudre ce jeu. Il est représenté par un grille de $M \times M$ cases et N couleurs où chaque case a une couleur générée aléatoirement au début de la partie.

Avant d'entrer dans les détails du jeu, on va d'abord introduire le terme "zone". Une zone est un ensemble où toutes les cases adjacentes (horizontalement et verticalement) ont la même couleur. La taille de chaque zone de la grille a une borne minimale de 1 (une zone qui contient une seule case), et une borne maximale de $M \times M$.

À chaque tour, le joueur va effectuer une opération de coloriage. Cette opération consiste à changer la couleur de toute la zone où se situe la cellule en haut à gauche. Cela a pour effet de relier la zone nouvellement colorée à toutes les zones voisines de cette couleur.

L'objectif général est de colorer totalement la grille en effectuant le moins de coloration possible. C'est-à-dire d'avoir qu'une seule zone à la fin.

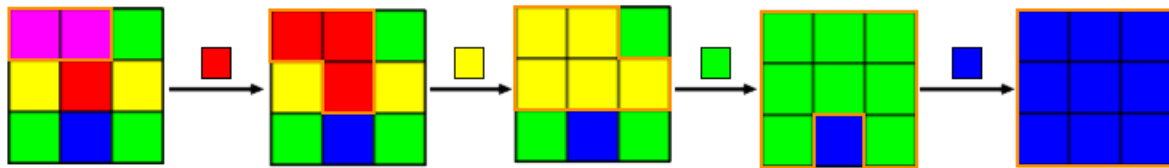


Figure 1.1 – Coloration d’une grille de 3x3

La figure 1.1, représente une séquence optimale de mouvements pour colorer une grille de 3x3.

1.3 Objectifs du projet

Ici, la tâche la plus compliqué n’est pas seulement d’implémenter des algorithmes permettant de résoudre Flood-It, mais d’implémenter ces algorithmes de tel façon à ce que l’on puisse les réutiliser sur d’autres casse-tête. Dès lors, plusieurs problématiques se posent :

- Comment appliquer des techniques de résolution dans le but de pouvoir les ré-appliquer sur d’autres casse-tête ?
- Comment modéliser notre programme pour faciliter l’implémentation d’un nouveau casse-tête ?
- Sachant que ce genre de casse tête sont des problèmes NP difficile, et que chercher à avoir une solution la plus proche que possible de la solution optimale peut s’avérer coûteux en temps et en espace, comment optimiser au maximum nos algorithmes ?
- Comment comparer l’efficacité des différents algorithmes ?

Maintenant que l’on a bien défini les objectifs de notre projet, on peut établir la liste des différentes tâches que nous devons réaliser :

- Modéliser et développer le jeu
- implémenter une interface visuelle du jeu
- développer les différentes méthodes de résolution
- faire un benchmark complet les différents algorithmes
- analyser et interpréter les résultats issu du benchmark
- bonus : implémenter un autre casse-tête sur lequel on va réutiliser des méthodes de résolution

A noter que parmi les différentes tâches que nous devons réaliser, il est impératif que le jeu, l’interface visuelle, et les différents algorithmes fonctionnent indépendamment. Cela est important pour la réutilisation. De plus, si les algorithmes et le jeu fonctionnent de façon dépendante, alors il sera impossible de pouvoir développer un autre casse-tête en utilisant des méthodes de résolution communes.

Chapitre 2

Planning

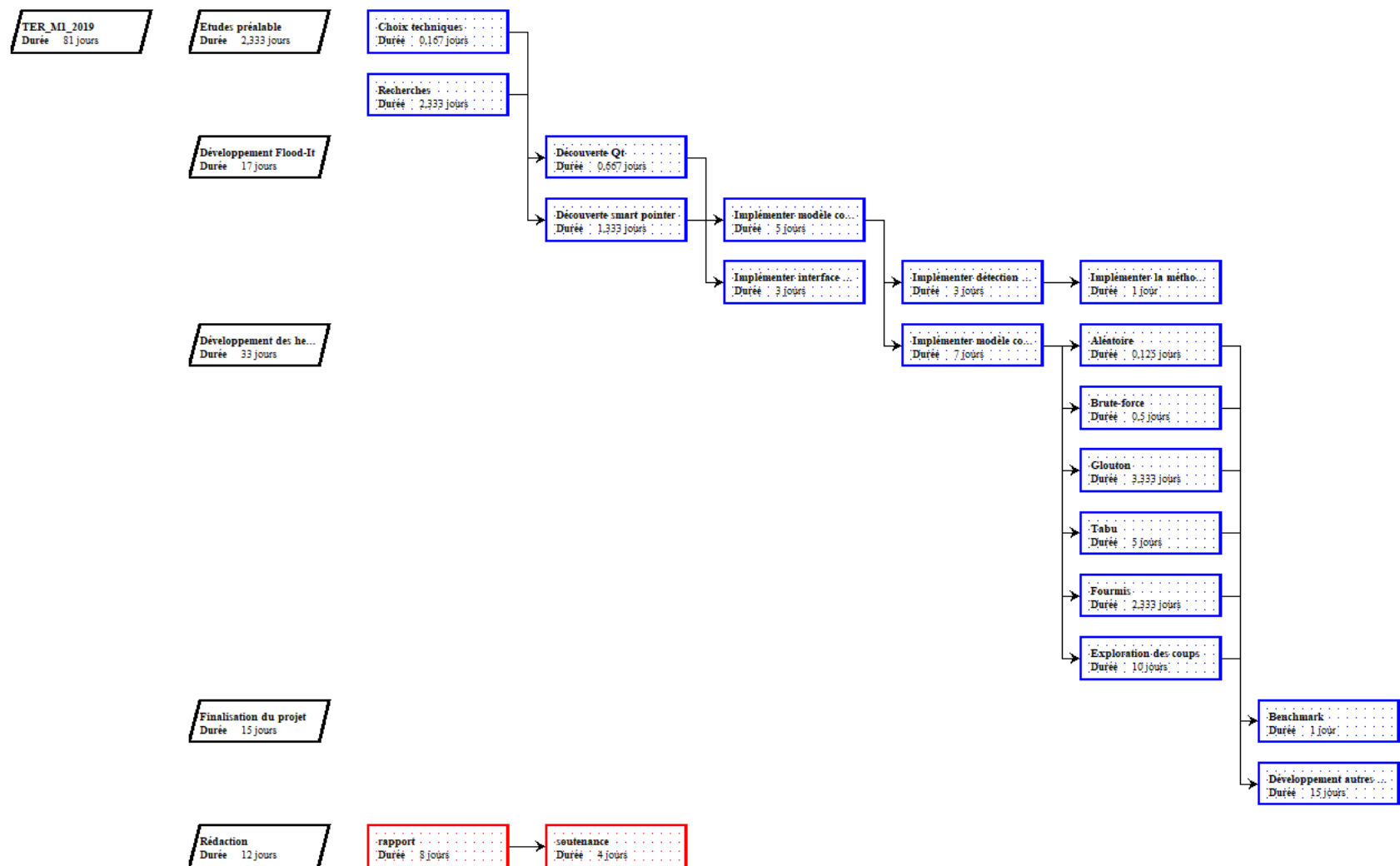


Figure 2.1 – Planning prévisionnel - Diagramme de Pert

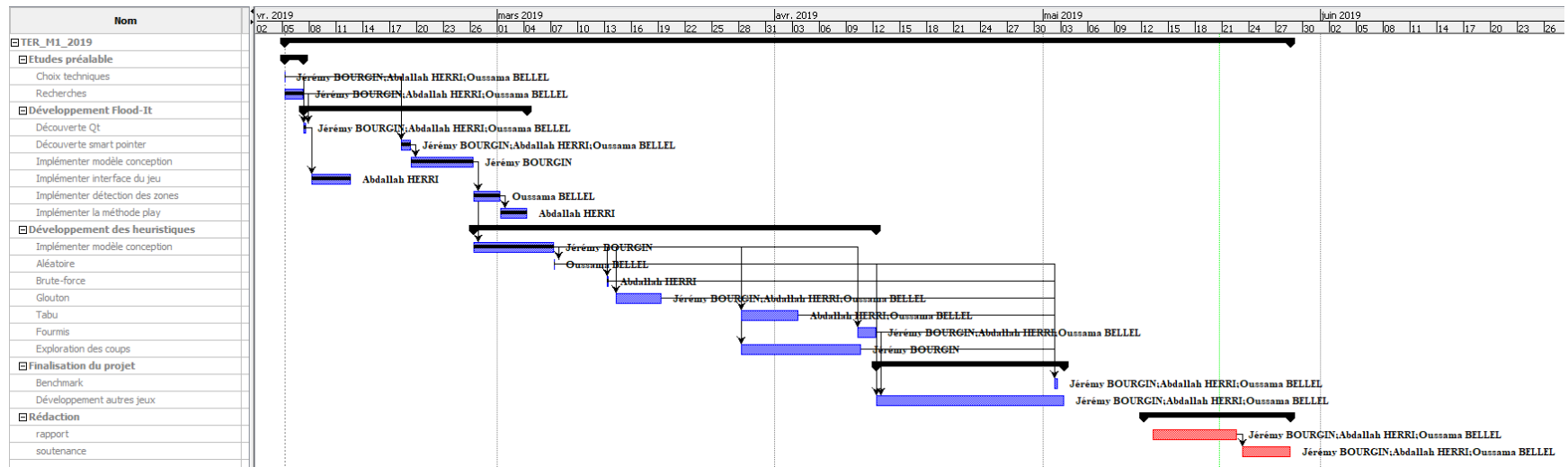


Figure 2.2 – Planning prévisionnel - Diagramme de Gantt

2.1 Planning final

Chapitre 3

Déroulement du projet

3.1 Données quantitatives

...

3.2 Données qualitatives

...

Chapitre 4

Conclusion

...