

RAPPORT PROJET C AVANCÉ



3A FISA
2023/2024

NOUTSA FOTSO Franck E. (TD 36)
RANA Baedaa (TD 36)
BRUN Clément (TD 34)

Table des matières

1	Introduction.....	3
1.1	Mots-clés	3
2	Contexte	3
3	Problématique	4
4	Conception et Méthodes	5
4.1	Structuration des Données.....	5
5	Fonctionnalités Implémentées.....	6
5.1	Création de la Forêt	6
5.2	Remplissage de la Forêt	6
5.3	Simulation d'Incendie	6
5.4	Voir l'état de la forêt à une itération spécifique	7
6	Outils Utilisés.....	7
6.1	Environnement de Développement Intégré (IDE)	7
6.2	Compilateur	7
6.3	Contrôle de Version	7
6.4	Gestion de Projet.....	8
6.5	Suite Microsoft Office	9
7	Résultats et Difficultés Rencontrées	10
7.1	Difficultés Rencontrées.....	10
7.2	Résultats	11
8	Conclusion	11
9	Bibliographie	12

1 Introduction

L'incendie de forêt (aussi connu sous le nom de feux de brousse, feux de broussailles ou feux de forêt) est un phénomène dévastateur qui peut entraîner des conséquences graves sur l'environnement, la biodiversité, et même la vie humaine. Incontrôlables et potentiellement destructeurs, ces grands incendies peuvent toucher les zones rurales et urbaines. Le vent peut se s'avérer être un allier redoutable car lorsqu'il transporte des braises et des étincelles, cela donne l'opportunité au feu de se propager à grande vitesse, de changer de direction et même « sauter » sur de grandes distances. (IFRC, s.d.)

1.1 Mots-clés

Simulation d'incendie – Feu – Forêt – Modélisation – Langage C – Programmation – Modulaire – Propagation – Gestion projet – Matrice – Développement

2 Contexte

Le projet de simulation d'incendie dans une forêt a été conçu dans le cadre d'approfondir nos connaissances de programmation en langage C. L'objectif principal de notre projet était de développer un programme en C capable de modéliser la propagation d'un incendie au sein d'une forêt, en utilisant des structures de données appropriées et en implémentant plusieurs fonctionnalités attendues et nos propres améliorations telles que le remplissage de la forêt (qu'il soit aléatoire ou manuel), la simulation de l'incendie, la modification des cellules (en changeant le type de la case par exemple, avec son état et/ou son degré), reprendre la simulation après ces modifications, la vérification de la propagation de l'incendie d'un point donné à un autre tout en calculant le chemin le plus court, l'ajout d'un menu attrayant pour lancer la simulation dès le début.

La modélisation de la propagation du feu dans une forêt revêt une importance cruciale pour la compréhension de ses mécanismes et la mise en place de stratégies de gestion et d'intervention. Ce type d'outil aide les pompiers dans leurs combats contre les flammes.

3 Problématique

Des milliers d'hectares de végétation sont ravagés chaque année par les incendies, causant aussi des dégâts humains importants et déplorables. La simulation d'un incendie au sein d'une forêt est une problématique pertinente, car elle permet de comprendre les mécanismes de propagation du feu et peut être appliquée à divers scénarios, tels que la gestion des incendies de forêt, la planification d'interventions, le sauvetage des civils pris entre les flammes et éventuellement mieux mesurer leurs conséquences. **(NADER, 2015)**

Nous avons exploré toutes les différentes étapes de la conception à l'implémentation de notre programme, en mettant l'accent sur le choix des algorithmes à utiliser, les structures de données, et avons rencontré plusieurs défis lors de notre processus de développement. L'objectif final que nous avons fixé à notre programme est qu'il puisse fournir une solution fonctionnelle et intuitive, avec une interface ludique et une expérience utilisateur agréable permettant de simuler et d'étudier la propagation d'un incendie dans diverses conditions simulées.

A travers ce rapport, nous partagerons notre démarche, nos résultats, et les leçons apprises au cours de ce projet captivant.

4 Conception et Méthodes

4.1 Structuration des Données

Pour réaliser ce projet, nous avons créé une structure de forêt, composée d'une grille de cellules, dans laquelle chacune des cellules représentent une unité de terrain dans la forêt. Chaque cellule a des propriétés telles que ;

- Le type : C'est un caractère représentant le type de la cellule. Chaque type a une signification spécifique (sol, arbre, feuille, etc.). Ceci permet de faciliter la gestion visuelle de notre forêt.
- L'état : Il est défini par un entier qui indique si la cellule est en feu [1] ou non [0].
- Le degré : C'est un entier positif qui représente le degré combustibilité de la cellule, permettant ainsi de modéliser la vitesse à laquelle le type peut brûler.

Notre structure de forêt permet à l'utilisateur de spécifier des dimensions de longueur et de largeur pour définir la taille de notre matrice. Pour un affichage visuel optimal en mode console (terminal), chaque type est symbolisé graphiquement à l'aide de symboles spécifiques, et de données spécifiques représentées dans le tableau suivant ;

Types	Symbole	Degré
SOL	+	0
ARBRE	*	4
FEUILLE		2
ROCHE	#	5
HERBE	x	3
EAU	/	0
CENDRES	-	1
CENDRES ETEINTES	@	0

Cette structure de données offre une représentation claire et modulaire de notre forêt, permettant ainsi une manipulation efficace des cellules et une gestion cohérente des informations nécessaires à la simulation d'incendie.

5 Fonctionnalités Implémentées

5.1 Création de la Forêt

La première fonctionnalité à implémentée dans le cadre de ce projet était la création de la forêt. Pour ce faire nous avons instancié une matrice de “cellules” avec ses attributs défini plus haut. Il a donc fallu faire beaucoup de recherche sur plusieurs points pour réaliser cette première étape, notamment sur les structures, les fichier headers aussi et surtout les pointeurs. Ce fut challengeant car au début de ce projet, aucun des membres du groupes n’était à l’aise avec le langage de programmation C et ses différentes spécificités.

5.2 Remplissage de la Forêt

Notre programme offre la possibilité de remplir la forêt de deux manières ;

- i. **Remplissage aléatoire** : Grace à cette option, chaque cellule est attribuée à un type spécifique (sol, arbre, feuille, roche, herbe, eau, on a exclu les cellules de types cendres et cendres éteintes ici) représenté par les symboles dédiés figurant dans le tableau ci-dessus. Cette fonctionnalité vise à créer des scénarios variés pour la simulation.
- ii. **Remplissage manuel** : Cette autre fonctionnalité permet à l'utilisateur de remplir manuellement chaque cellule de la forêt en spécifiant son type. Cela offre une fine personnalisation de la composition de la forêt, permettant à l'utilisateur de définir des configurations spécifiques pour la modélisation.

5.3 Simulation d'Incendie

Cette fonctionnalité a été soigneusement implémentée lors de l’écriture de notre code source. Lors de l’utilisation de cette fonctionnalité, l’utilisateur entre le nombre d'itérations de la simulation, puis lance la simulation de propagation du feu et affiche l'état de la forêt à chaque itération. Dans notre programme, un incendie peut se propager d'une cellule à une autre en fonction du degré de chaque type. Dans ce contexte, nous comprenons donc qu’une feuille avec un degré de 2, brûlera plus rapidement qu'un arbre, dont le degré est de 4.

De plus, notre simulation permet à l'utilisateur de déclencher l'incendie à une position spécifique de son choix (en choisissant à l'aide d'un curseur la cellule à embraser).

5.4 Voir l'état de la forêt à une itération spécifique

Nous avons également implémenté une fonctionnalité permettant de stocker les états de la forêt dans un tableau tridimensionnel afin de permettre à l'utilisateur de visualiser à la fin de la propagation du feu l'état de la forêt à une itération donnée.

6 Outils Utilisés

Tout au long de notre projet de simulation d'un incendie dans une forêt en langage C, plusieurs outils ont été utilisés pour réaliser au mieux notre projet. Voici une liste d'outils utilisés :

6.1 Environnement de Développement Intégré (IDE)

Nous avons utilisé l'IDE CodeBlocks et l'éditeur de texte Visual Studio Code (en installant le plugin C/C++) pour l'écriture de notre code source.

6.2 Compilateur

Nous avons utilisé GCC (GNU Compiler Collection) pour la compilation de notre code source en langage C. GCC est une collection de logiciels open-source (libres) complètement intégrés et capables de compiler divers langages de programmation, dont C. Il a été créé par le projet GNU. **(GCC, 2013)**

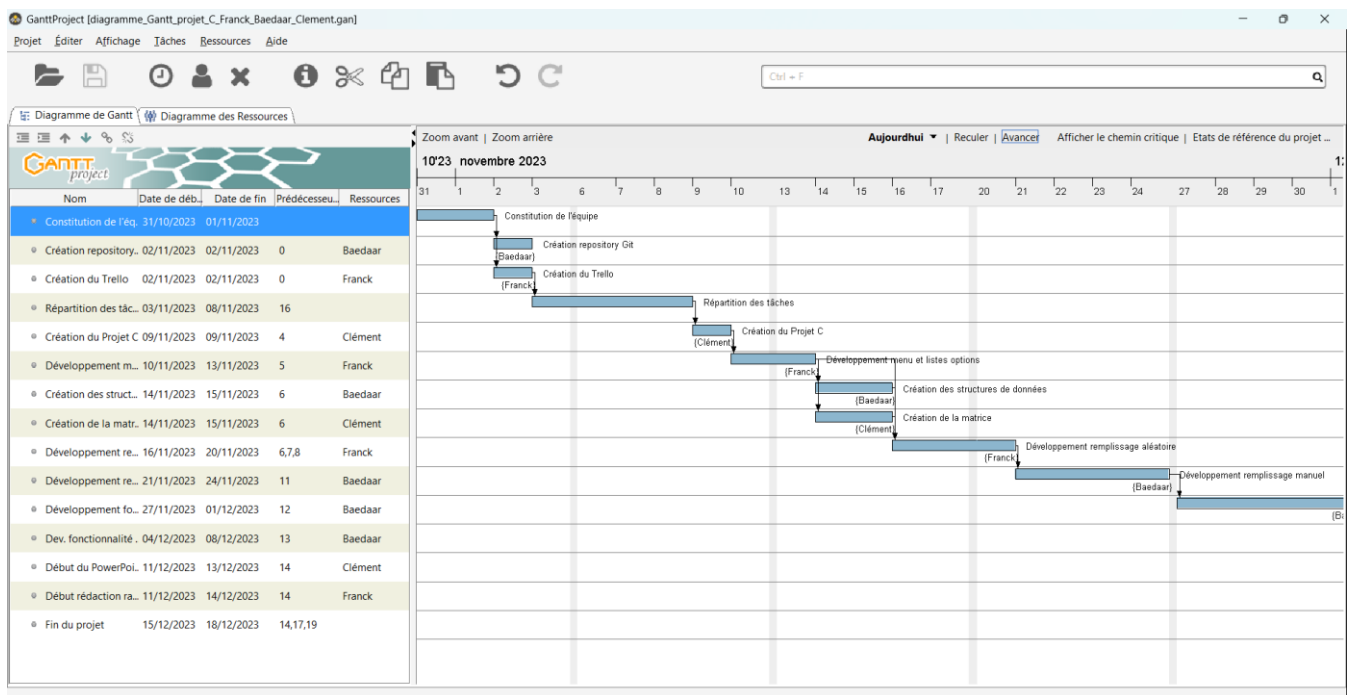
6.3 Contrôle de Version

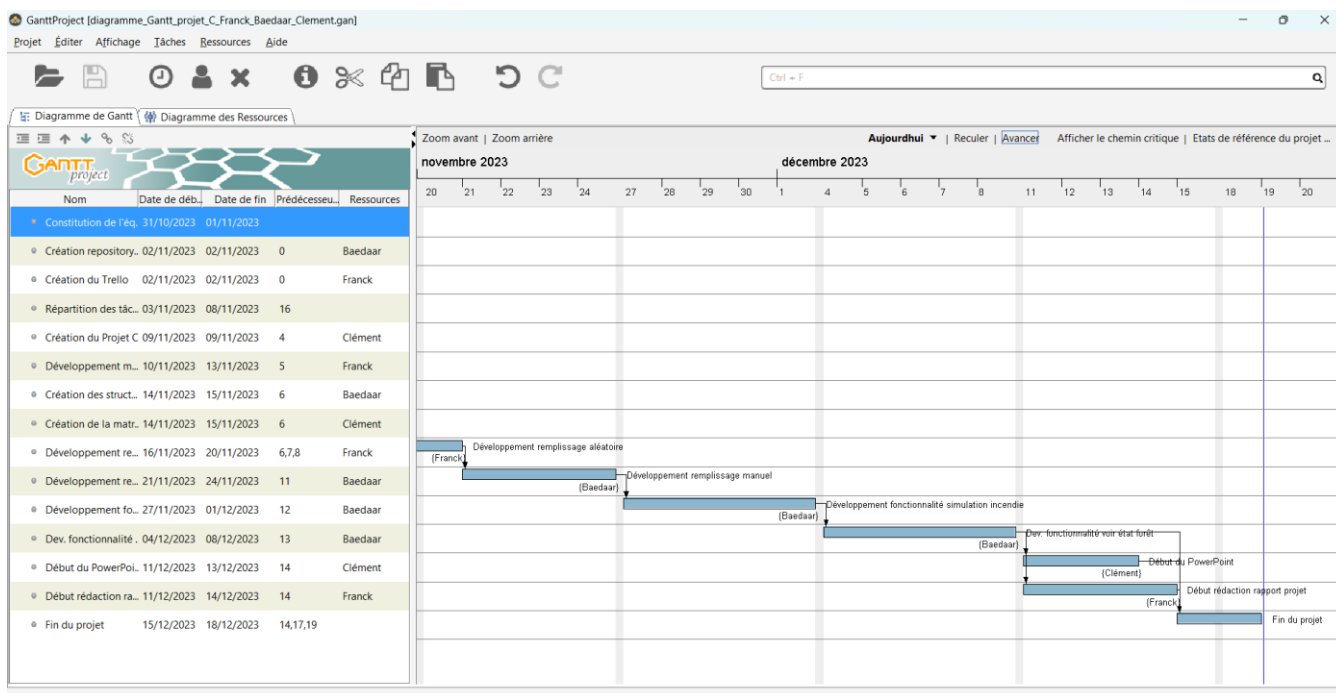
Pour le suivi des versions du code source, la collaboration entre les membres de l'équipe, nous avons utilisé Git qui est un fabuleux outil, nous avons également utilisés GitHub, plateforme de gestion de code source basée sur Git pour héberger notre code pour y avoir accès partout et faciliter le suivi des problèmes.

6.4 Gestion de Projet

Ici nous allons mettre en évidence les outils spécifiquement utilisés pour la gestion du projet.

- i. **Trello** : Principalement utilisé pour organiser et suivre les tâches, assigner des responsabilités aux membres de l'équipe, et visualiser l'état d'avancement du projet, Trello est un outil de gestion de projet performant basé sur des tableaux Kanban.
- ii. **GanttProject** : C'est un outil utilisé pour créer des diagrammes de Gantt permettant de planifier et suivre les échéanciers des différentes phases du projet. La gestion du temps est largement simplifiée grâce à cet outil.





6.5 Suite Microsoft Office

Nous avons principalement utilisé Microsoft Word pour la rédaction et la mise en forme de ce rapport de projet, et Microsoft PowerPoint pour la création et la préparation des diapositives pour la présentation visuelle devant notre enseignante lors de notre défense. Ceci nous permettra de soutenir efficacement notre narration orale.

Pour le bon fonctionnement de ce projet, nous avons utilisé une multitude de Bibliothèques C qui font partie des plus couramment utilisés telles que les bibliothèques « **stdio.h** », « **stdlib.h** » et pleins d'autres. Nous avons aussi utilisé des bibliothèques tierces telles que Ncurses pour des fonctionnalités spécifiques à l'affichage, comme énoncé plus haut.

7 Résultats et Difficultés Rencontrées

7.1 Difficultés Rencontrées

Lors de la réalisation de notre projet, nous avons rencontré pas mal de défis qui ont su nous la ralentir dans notre progression. Par exemple ; lors de la gestion de la mémoire dynamique, en particulier lors de l'allocation et de la libération des ressources pour la grille de la forêt, ou pour tout type complexe en général dans le projet.

L'implémentation de la simulation d'incendie a également nécessité une attention particulière pour garantir la précision du modèle, nous avons d'abord essayé une approche récursive mais on s'est rendu compte qu'il était plus simple d'approcher ce problème de manière itérative sans récursivité, il serait toute fois intéressant de réessayer l'approche récursive dans une version ultérieure.

Un des problèmes majeurs que nous avons rencontrés a été la gestion des pointeurs, en effet nos connaissances en C et par conséquent en pointeurs étaient très limitées, nous avons dû faire beaucoup de recherches et essayer beaucoup de chose pour arriver à quelque chose de fonctionnel étant donné qu'il a fallu utiliser des "doubles" et "triples" pointeurs et ne pas se limiter aux simples, en tout cas c'est comme ça que nous avons pu avancer.

Concernant le technique un soucis que nous avons rencontrés a été le manque de compatibilité des bibliothèques sur les différents os de l'équipe, en effet en ce qui concerne la bibliothèque « Ncurses » nous l'utilisons pour gérer l'utilisation des touches directionnelles, le curseur et l'affichage de la forêt dans le projet, le problème étant que cette librairie ne fonctionne que sur les systèmes Unix et mac (même noyau), pour pallier à ça nous avons souvent du travailler en pair programming pour avancer.

Note ; Ncurses n'est pas disponible sur Windows. C'est une bibliothèque couramment utilisée sur les systèmes basés sur Unix, tel que Linux.

Le fait que notre groupe soit composé d'étudiant de TD différents a également été contraignant mais nous avons pu passer outre grâce aux outils de gestions utilisés lors de ce projet.

7.2 Résultats

Malgré les nombreuses difficultés rencontrées, nous avons finalement pu obtenir un programme agréablement fonctionnel capable de simuler la propagation d'un incendie dans une forêt.

8 Conclusion

Nous arrivons à la fin de ce rapport. La réalisation de ce projet C avancé sur la simulation d'un incendie en forêt, nous a permis de consolider nos connaissances respectives en programmation en langage C, d'apprendre davantage sur la programmation en général et d'appliquer des concepts importants tels que les structures de données, la gestion de la mémoire, les pointeurs, et la modélisation d'algorithmes. La simulation d'un incendie dans une forêt est un sujet fascinant qui a ajouté une dimension pratique à notre apprentissage.

En conclusion, ce projet nous a fourni une expérience enrichissante et a grandement renforcé notre compréhension des concepts de programmation.

9 Bibliographie

GCC. (2013, Juin 19). Récupéré sur

[https://dms.umontreal.ca/wiki/index.php?title=Accueil:](https://dms.umontreal.ca/wiki/index.php?title=Accueil)

[https://dms.umontreal.ca/wiki/index.php/GCC#:~:text=GCC%2C%20abr%C3%A9viation%20de%20GNU%20Compiler,%2C%20Java%2C%20Ada%20et%20Fortra](https://dms.umontreal.ca/wiki/index.php/GCC#:~:text=GCC%2C%20abr%C3%A9viation%20de%20GNU%20Compiler,%2C%20Java%2C%20Ada%20et%20Fortran.)
n.

Gerçeker, R. K. (2005, Janvier 14). *Introduction à Ncurses*. Récupéré sur

<http://www.linuxfocus.org>: http://www.linuxfocus.org/Francais//Archives/If-2002_03-0233.pdf

IFRC. (s.d.). *Les incendies de forêt*. Récupéré sur www.ifrc.org:

<https://www.ifrc.org/fr/notre-travail/catastrophes-climat-et-crises/quest-ce-quune-catastrophe/les-incendies-foret>

NADER, B. (2015, Juin 19). *Evaluation des simulations de feux de forêts*. Récupéré sur

[hal.science: https://theses.hal.science/tel-01403922v1/file/pdf2star-1450107568-These_NADER_Bahaa-VF.pdf](https://theses.hal.science/tel-01403922v1/file/pdf2star-1450107568-These_NADER_Bahaa-VF.pdf)