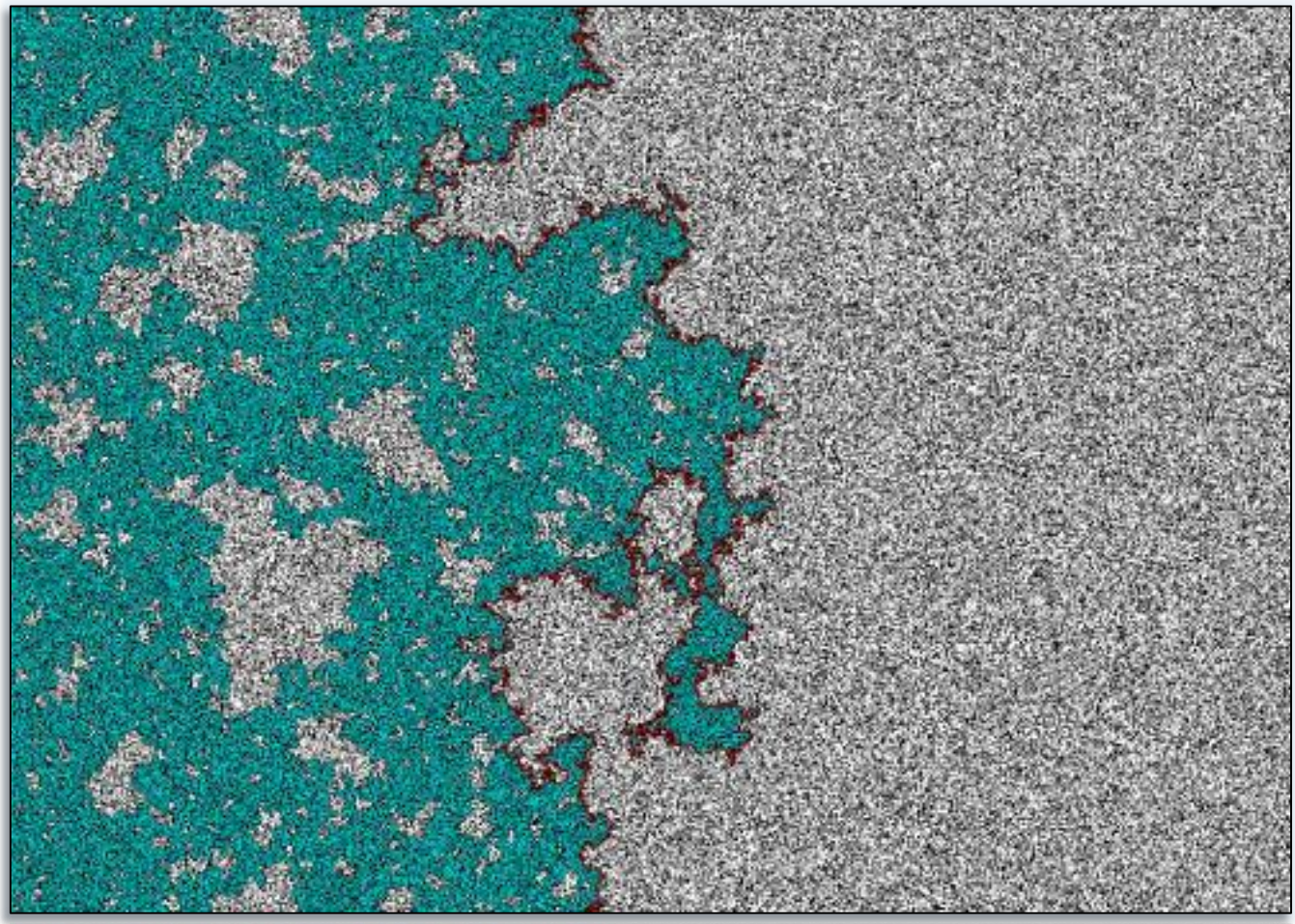
Rapport de synthèse:

Simulation d'un feu de forêt

(Cas 4)



Mai – Juin 2020

Je devais dans ce projet produire une application de jeu visant à simuler un feu de forêt en langage java, bien que les incendies ne soient pas si drôles. J'ai donc étudié la propagation des incendies, via des modèles mathématiques existants, mais ai aussi étudié des simulations similaires à ce qui m'a été demandé. Évidemment, le travail final visé reste à mon échelle d'étudiant, mais il faut savoir que de vrais logiciels de simulations d'incendies [1] sont utilisés par des professionnels, afin d'anticiper et de comprendre les déplacements du feu. Ce rapport présente la synthèse de mon travail.

Plan d'étude

I. Présentation du projet

A. Contexte

« Un incendie est un feu non maîtrisé, ni dans le temps, ni dans l'espace. La caractéristique d'un incendie est de pouvoir s'étendre rapidement et occasionner des dégâts généralement importants. Ses conséquences sont destructrices tant sur l'environnement dans lequel il évolue que sur les êtres vivants qu'il rencontre. » [2].

La structure des incendies est connue mais les paramètres extérieurs peuvent réellement compliquer sa gestion: par exemple la végétation, la densité de la végétation, le climat ou encore le vent.

Mathématiquement, le phénomène est bien connu. On parle d'un phénomène de percolation: la théorie de la percolation a été introduite par J.Hammersley en 1957 alors qu'il prenait un café. En effet, cette théorie donne un comportement aux milieux aléatoires et elle est aussi utilisée dans la physique des matériaux pour comprendre les déplacements des fluides en leur sein, puis par analogie, aux feux de forêts. Cette théorie se base sur le concept de graphe aléatoire dont le travail en bibliographie donne un rapide aperçu [3].

Dans ce projet, le seul paramètre que l'utilisateur pourra contrôler sera la densité d'arbres de la forêt qui lui permettra d'observer le phénomène de percolation suivant qu'on dépasse le seuil ou pas (cf. phénomène de percolation).

Les travaux suivants, réalisés en langage Python, donnent un aperçu intéressant de ce phénomène [3] ; que je vous invite à consulter.

B. Hypothèses

Les hypothèses qui ont servi à simplifier le modèle sont les suivantes :

- pas de temps et d'espace discret i.e discrétiser un phénomène continu en incrémentant le temps et l'espace (utilisation de tableau et de grille),
- la densité de la forêt est un paramètre de notre modèle comprise entre 0 et 1 exclus,
- considérons un arbre (1 case de la grille), si un de ses 4 voisins dans les directions cardinales est en flamme, alors il le sera aussi (arbre = récepteur),
- et pendant un seul incrément de temps (combustion complète en 1 pas de temps),

Ces hypothèses m'ont permis de cadrer le développement, et peuvent être, dans une amélioration future, revues.

C. Contexte de programmation

J'ai codé cette application en langage Java. Ce langage offre une bibliothèque graphique que j'ai utilisé pour gérer l'affichage : SWING. Et bien-sûr, la bibliothèque AWT, qui est l'ancêtre de SWING sur lequel il repose toujours.

II. La modélisation finale

Voici, le diagramme de cas d'utilisation qui n'a pas changé (ci-dessous). L'utilisateur, lorsqu'il lance la simulation veut pouvoir contrôler le paramètre de densité, et que ce soit visuel : voir la propagation.

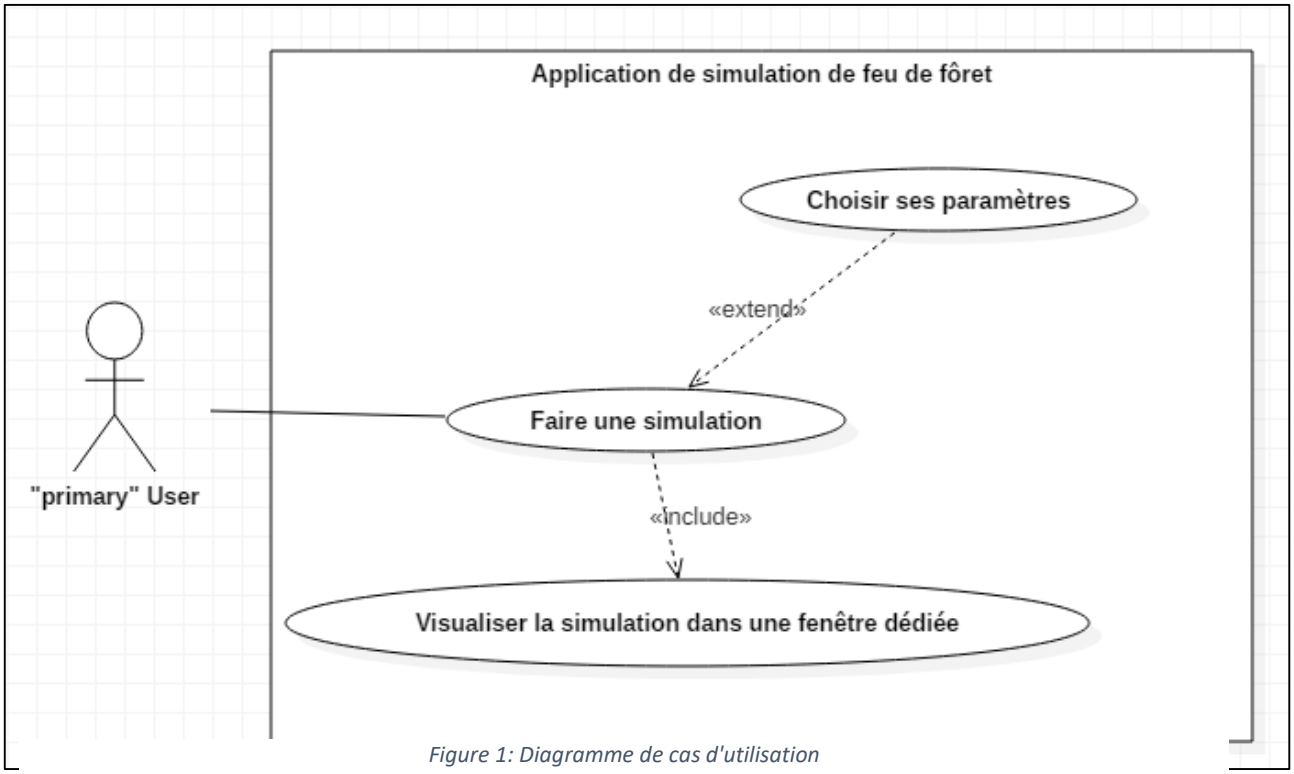


Figure 1: Diagramme de cas d'utilisation

Le diagramme de classe lui a quelques peu évolué notamment certaine classe comme Forêt dont les attributs sont passés statiques. Cela m'a donné la possibilité de créer un unique objet Forêt, que j'ai appelé dans les autres classes.

J'ai inséré une classe Grille, qui n'était pas sur mon le précédent, qui m'a permis de gérer l'affichage indépendamment de la classe Forêt pour ne pas la surcharger. Les classes commençant par « initiale » servent à initialiser les conditions initiales, notamment la densité, rentrée par l'utilisateur.

Sur une suggestion extérieure, j'ai modifié la relation entre les trois classes principales : Forêt, Feu et Propagation ; en mettant une relation d'association, qui était plus adaptée.

Je me dois aussi citer le travail de Pierre Audibert, professeur agrégé de l'Enseignement Supérieur, et Ingénieur des Ponts et Chaussées, pour son travail [4] qui m'a inspiré sur le processus et sur la structure de certaines méthodes.

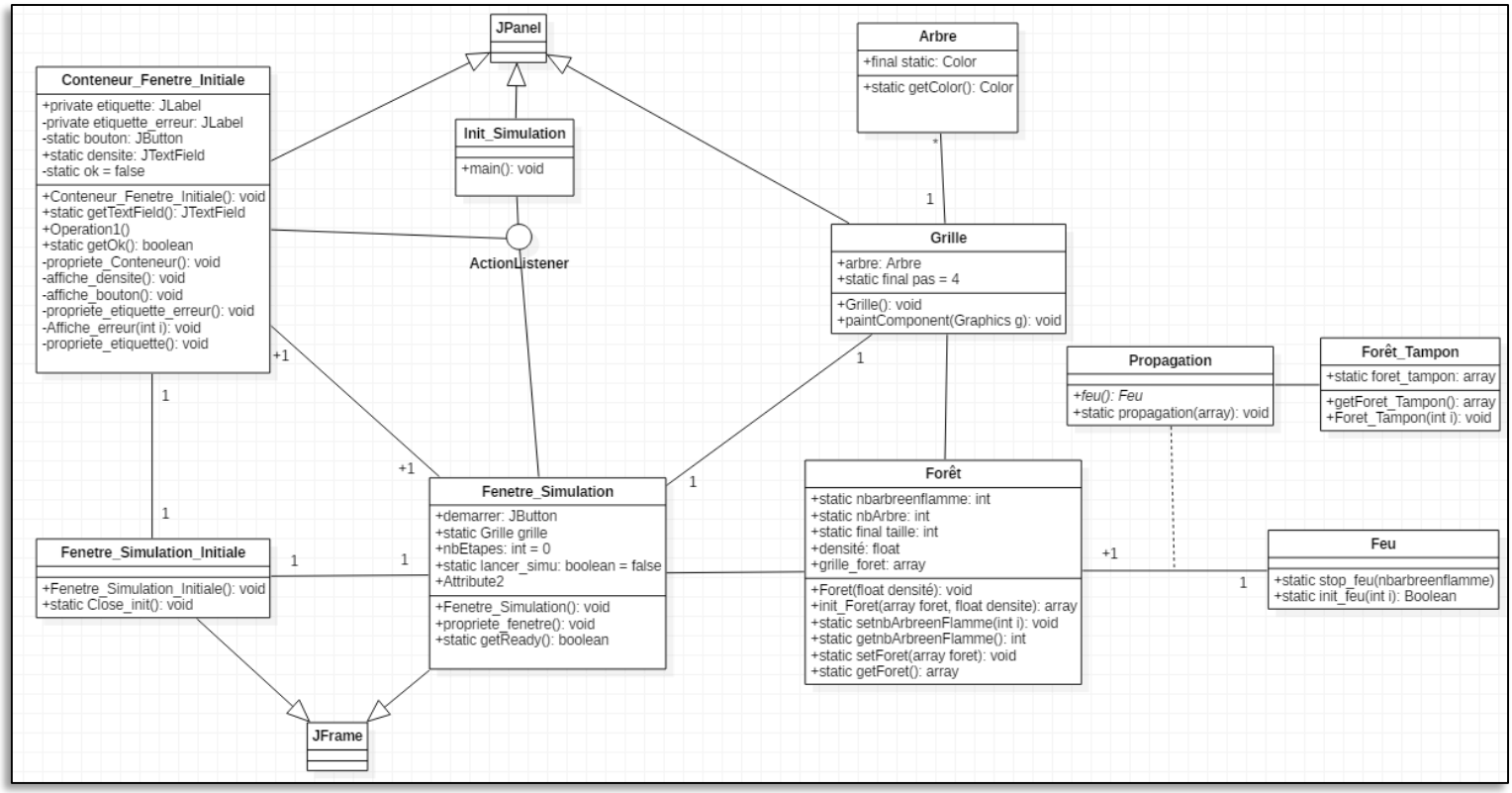


Figure 2: Diagramme de Classe

Les diagrammes d'activité n'ont pas évolués, je les ai joins en Annexe 1.

Ils m'ont guidé à travers le développement. Surtout de la fonction principale de la simulation : la propagation du feu.

Concernant le diagramme de séquence, je l'ai fait évoluer, notamment sur la condition d'arrêt de la simulation, où j'ai changé un « if » contre un « do … while », que j'ai trouvé plus pertinent pour gérer la mise à jour.

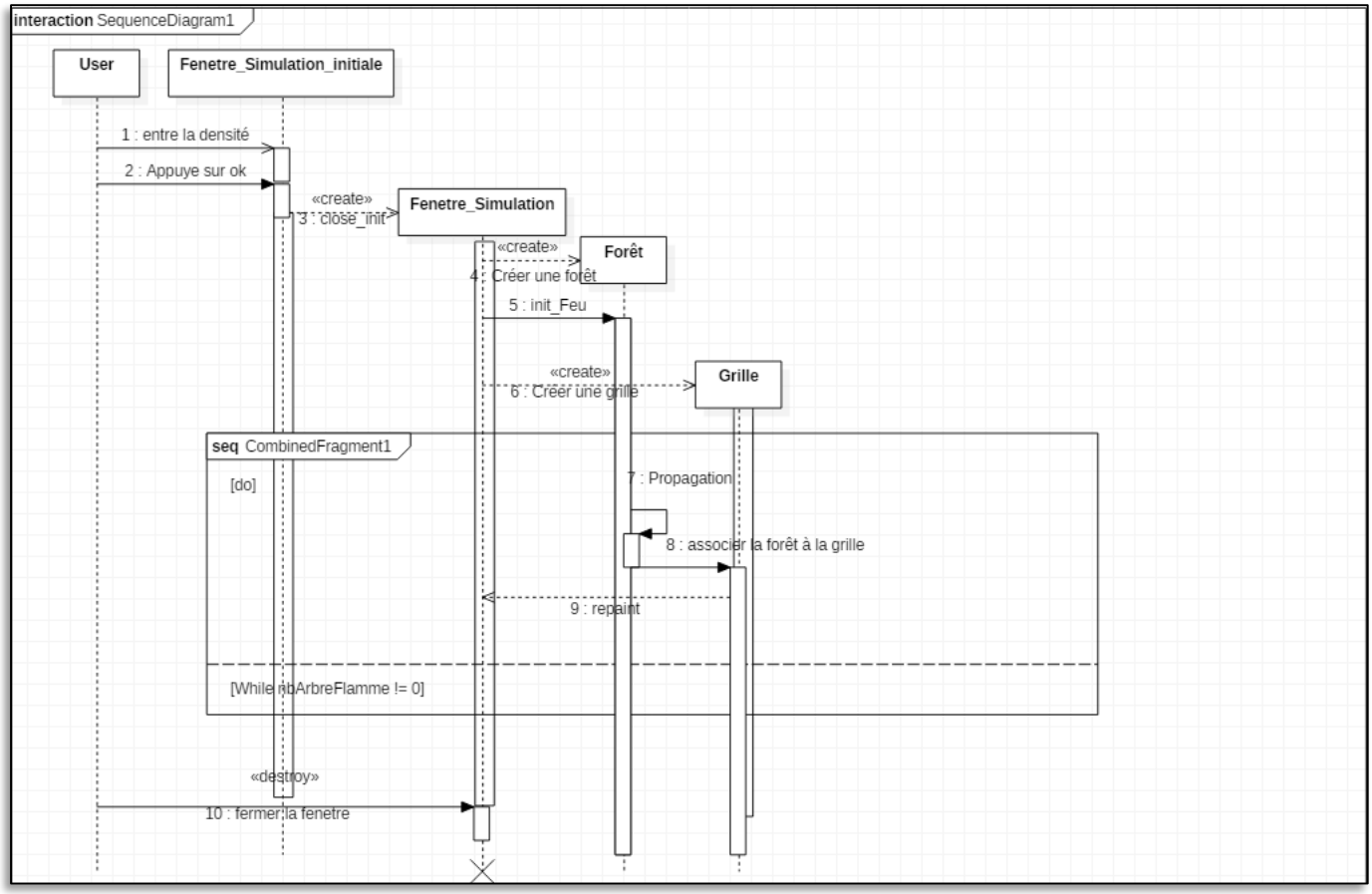


Figure 3: Diagramme de Séquence

III. Le Déroulement du projet

Étant simultanément en stage et sur ce projet, la ressource de temps a été précieuse. Le déroulement du code a pris plus de temps que prévu, ce que j'avais anticipé dans mon diagramme de GANTT lors de la phase d'analyse. J'ai ainsi pu finir dans les temps ce qui m'a été demandé. Cependant, le temps m'a manqué pour améliorer mon modèle en requalifiant et réévaluant certaines hypothèses.

Le travail de recherche a constitué une grande partie de mon travail, notamment sur le fonctionnement propre des fonctionnalités de Java. Un des problèmes auquel j'ai été confronté est le fait que Swing n'est pas « thread safe » ; ce qui implique de créer des «thread» pour ne pas provoquer de blocage de l'«EventDispatchThread». Son fonctionnement, s'il n'est pas connu, rend les affichages trop lents. C'est ce qui m'a bloqué pour afficher de manière interactive la propagation. Les états initial et final s'affichent mais pas les étapes intermédiaires, ce qui pose problème pour le côté visuel. Je pense que l'erreur vient de mon appel de la classe Fenetre_Simulation.

IV. Conclusion

Ce projet m'a permis de revoir les notions essentielles du langage Java et d'affiner ma compréhension de ses mécanismes. Bien manipulé, c'est un outil puissant qui a une grande capacité d'adaptation à n'importe quel problème grâce à son potentiel d'abstraction.

J'aurai souhaité aller plus loin mais le temps m'a manqué. J'aurai notamment voulu obtenir un affichage progressif.

Pour améliorer cette application, plusieurs pistes sont apportées par P.Audibert dans son travail, notamment le fait que les arbres ne se consument pas en un seul pas de temps mais progressivement. Nous pourrions très bien, ajouter d'autres paramètres que la densité, ou permettre à l'utilisateur de choisir un type de forêt bien précis, ainsi que le climat. L'utilisateur pourrait simuler sous le climat qui l'intéresse, et pourquoi ne pas, ajouter du relief et du vent, afin de se rapprocher de la réalité. Le phénomène de percolation est aussi utilisé en épidémiologie, où ce n'est pas le feu qui se transmets mais une maladie. Nous pouvons trouver d'autres applications de cette simulation et l'adapter graphiquement.

V. Bibliographie

[1] Simulateur de feux de forêts, Sapeurs-Pompiers du Var, https://www.sdis83.fr/internet/simulateur-feux-de-forets.html.

[2] Incendie, Wikipédia, https://fr.wikipedia.org/wiki/Incendie

[3] Murielle Jarry et Pierre Campet, Modélisation d'un feu de forêt, http://www.mathom.fr/mathom/sauvageot/Modelisation/Graphes/Feux_foret.pdf, Juin 2010

[4] Pierre Audibert, Simulation d'un feu de forêt, http://www.pierreaudibert.fr/tra/feudeforet.pdf

Annexe 1

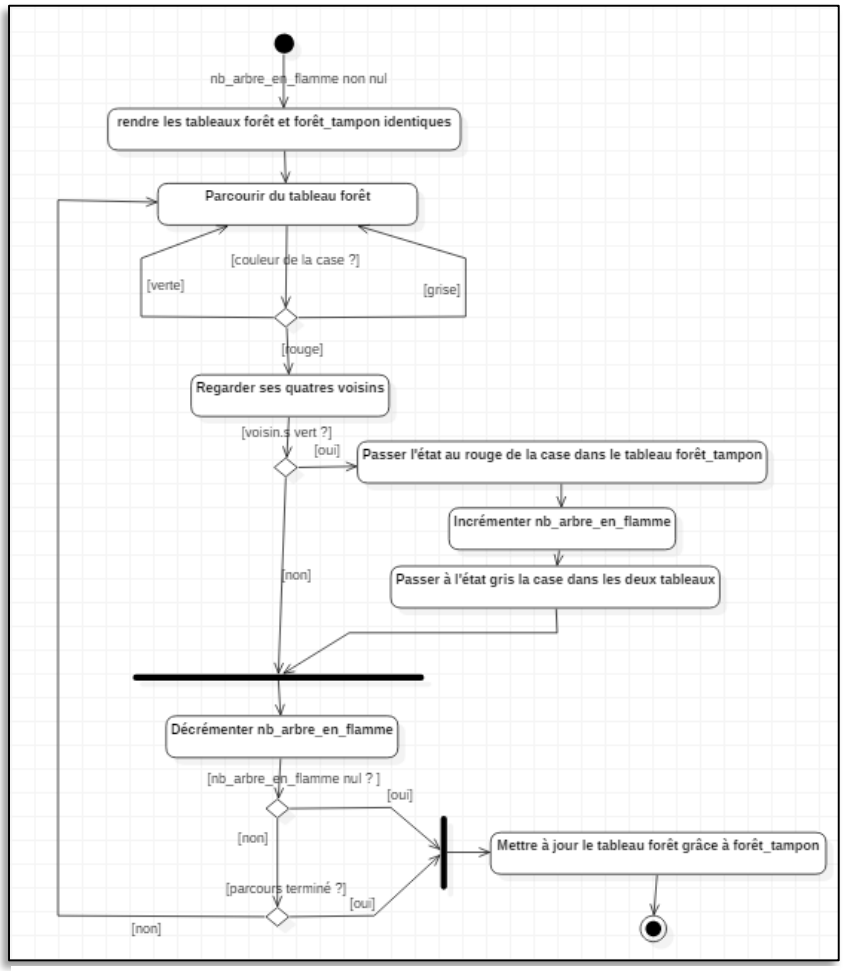


Figure 4: Diagramme d'activité de la fonction "Propagation"

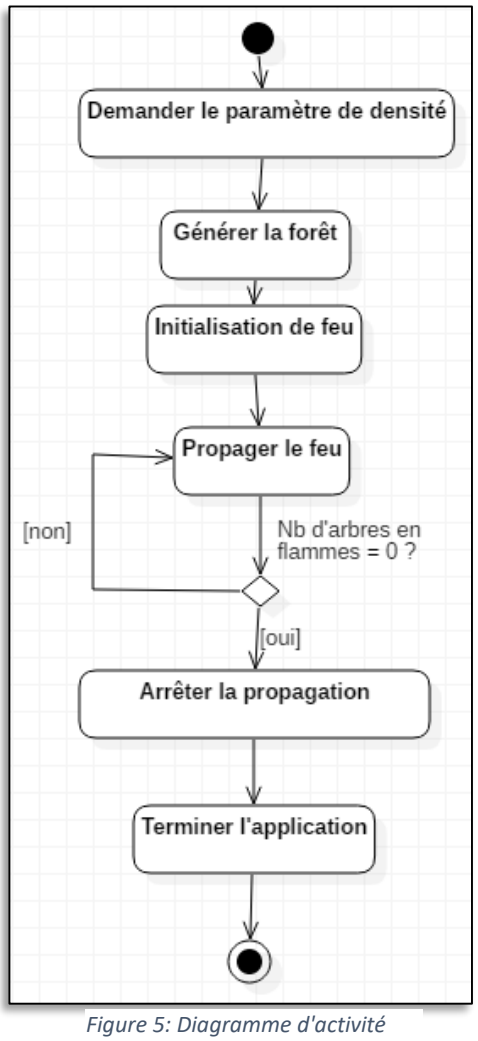


Figure 5: Diagramme d'activité