



ÉCOLE  
**D'INGÉNIEURS**  
PARIS-LA DÉFENSE

# PROJET INCENDIE

**SIMULATION DE LA  
PROPAGATION  
D'UN INCENDIE DE FORÊT**

ALGORITHMIQUE  
ET LANGAGE C#

**RAPPORT DE PROJET 1<sup>RE</sup> ANNÉE RESTART 2020**

Karolina KLAK, Héloïse FOURDAIN et Ilies GOURRI

# SOMMAIRE

## PRÉSENTATION DU PROJET

### 1. Objectifs

- Objectifs de l'équipe
- Objectifs du projet

### 1. Objectifs

- Objectifs de l'équipe
- Objectifs du projet

### 2. Déroulement

- Compréhension et assimilation du sujet
- Avoir à programmer avec plusieurs fichiers
- La conception des méthodes
- L'organisation du code
- Le fichier de lecture
- Travailler à distance

### 3. Explication de méthodes

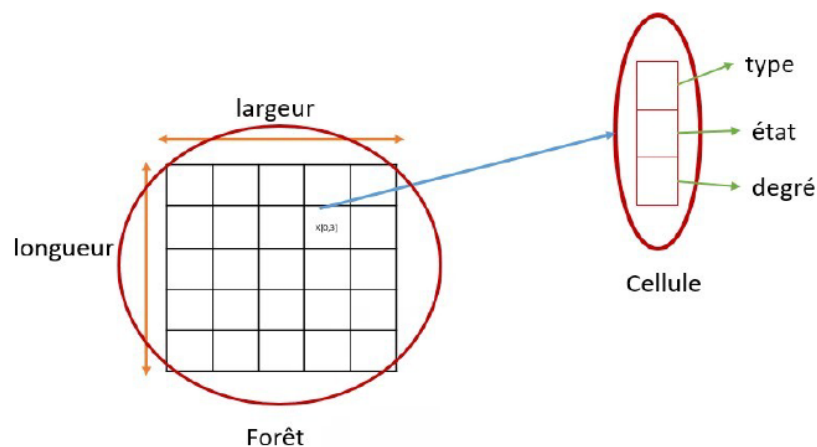
- Cellule
- Program
  - Explication InitialiserFeu
  - Explication NbVoisinsFeu

### 4. Contribution de chacun

## CONCLUSION

# PRÉSENTATION DU PROJET

*Le but du projet est de simuler la propagation d'un incendie de forêt grâce aux différents outils dont on dispose grâce à notre formation « Algorithmique et langage C# ». En effet, à l'aide à l'utilisation d'une matrice on peut assimiler chaque cellule à un élément de cette forêt. Le jeu consiste à incendier aléatoirement un élément de la forêt (dont on choisit la taille) puis de choisir un nombre tour qui correspond à la durer de l'incendie. Le jeu nous renvoie finalement une forêt plus ou moins brûlée selon la taille et le nombre de jours choisis.*



# 1. Objectifs

## • Objectifs de l'équipe

Sur un plan personnel, et éducatif, le projet nous a permis de vérifier si les connaissances censées être acquises lors des cours et des TDs étaient bien assimilées et comprises.

Nous avons pour objectif d'apprendre à travailler en coopération sur du même code, étant donné que les idées divergent, cela semblait risqué, et sibyllin, l'équipe devant adopter une communication rigoureuse.

Un autre objectif important pour nous a été d'apprendre à bien s'organiser en groupe. Nous avons déjà travaillé ensemble sur le PIX1, mais le projet en algorithmique nous a permis d'affirmer notre écoute au sein de l'équipe, le tout reposait en effet sur du codage, quelque chose de nouveau cette année pour tous les membres du groupe.

## • Objectifs du projet

Le projet a eu pour but avant tout de servir à l'apprentissage de chacun du langage C#, en gérant un projet donné. Ainsi, on a pu améliorer notre organisation, et nos efficacités individuelle et collective. Il avait pour but de nous permettre d'assimiler les notions apprises en cours, et travaillées lors des TD.

Par ailleurs, le projet avait pour objectif de développer notre logique mathématique nécessaire en programmation.

# 2. Déroulement

## • Compréhension et assimilation du sujet

À la première lecture du sujet du projet, il nous a fallu du temps pour comprendre ce qui était demandé. En effet il s'agit de notre premier devoir à rendre en informatique. De plus, les consignes semblaient denses et compliquées à comprendre.

Avec du temps et de la patience, notre équipe a pu appréhender de manière efficace cette consigne qui nous semblait ardue.

Néanmoins, certaines parties de l'énoncé étaient assez floues ou illogiques, notamment la propagation de l'incendie : que se passe-t-il donc lorsque la case  $[i, j]$  est une feuille ? Nous avons donc convenu que ce n'était qu'un oubli, et que la propagation par le biais d'une feuille n'était pas possible.

### • **Avoir à programmer avec plusieurs fichiers**

Cette nouveauté nous a légèrement perturbés mais à l'aide de camarades de classe nous avons pu comprendre rapidement les subtilités du projet. Les constructeurs nous ont posé problème notamment celui utilisé dans LireFichier par son assignation de string en fonction du type que l'on a passé en paramètre. De plus, il nous a été difficile de désigner le fichier de destination des différents éléments du code au début.

### • **La conception des méthodes**

Nous nous sommes reposés premièrement sur l'organisation proposée dans le rapport puis nous nous sommes très vite rendu compte que pour la clarté et pour l'esthétisme du code nous devons concevoir des méthodes supplémentaires à celles proposées dans le rapport. La compréhension de l'algorithme nécessaire au bon fonctionnement de la méthode et le lien entre ces méthodes fut rétif. Afin de résoudre cet inconvénient, nous avons adopté avec le groupe entier la méthode du papier-crayon ainsi que du brainstorming, que l'on enregistrerait et que l'on pourrait réécouter par la suite pour se souvenir de ce qui avait été dit. Nous faisons face à une erreur, `System.IndexOutOfRangeException`, elle se trouve vraisemblablement dans la méthode `NbVoisinsFeu`. Elle est la raison pour laquelle le lancement du programme s'arrête après le premier affichage de la matrice. Nous conseillons donc d'essayer de lancer plusieurs fois le programme pour observer l'effet 666 sur la forêt.

### • **L'organisation du code**

Avec la gestion de la classe `Cellule` et du `Program`, nous devons donc réfléchir à une organisation des méthodes logique et efficace. Ainsi, une fois les méthodes à utiliser comprises, nous avons pu déterminer comment organiser le code. Néanmoins, ce n'était qu'une illusion de début de projet, puisque très vite, nous nous sommes retrouvés avec des méthodes incorrectes, dont le résultat n'était pas celui recherché. Nous avons alors dû réorganiser plusieurs fois notre code. Finalement cette organisation a été comprise au fil de l'écriture, puisque c'est ainsi que nous avons pu au mieux comprendre

le fonctionnement de chaque méthode et la cohésion nécessaire de chacune avec les autres. À la fin de l'écriture de toutes les méthodes nous avons encore dû repenser l'organisation, puisqu'il fallait rendre l'écriture un peu plus lisible, esthétique.

### • **Le fichier de lecture**

On cherche à lire le fichier, pour ce faire on crée une méthode. Un moyen de connaître le nombre de lignes et de colonnes a été de les écrire puis de faire un `ReadLine()` pour les enregistrer. On a stocké des variables du même type, ici les types des cases de la forêt, dans un même tableau, avec `data[k]`. On a dû d'abord le définir, puis après avoir séparé le `data[k]` grâce à un élément séparateur, nous avons créé toutes les cellules.

La création du fichier nous a permis de stocker les informations de manière persistante dans notre cas.

### • **Travailler à distance**

Le contexte actuel nous a obligés à coopérer à distance, il était donc nécessaire de s'adapter et de trouver des solutions.

De plus, le projet était avant tout un travail d'équipe. Le principal souci a été la communication de nos idées, et de nos principes. Chacun a une logique et une manière de coder différentes. Il a donc fallu coopérer et savoir se remettre en question afin de garder la bonne cohésion du groupe. Nous avons malheureusement dû faire face aux problèmes liés à la connexion internet, qui rendaient irascibles les séances de brainstorming collectif, les moments où l'on s'expliquait différentes parties, ou que l'on réfléchissait ensemble à des solutions aux endroits où l'on bloquait. En effet, ces moments étaient souvent accompagnés de partage d'écran, ainsi la mauvaise connexion rendait les échanges difficiles et frustrants. Mais nous avons toujours réussi à surmonter cela, et garder calme en toutes circonstances.

Pour avancer de manière la plus efficace, nous avons partagé les différentes méthodes, contributions de chacun par Teams, application qui nous a aussi permis de partager l'écran en cas d'erreurs farfelues.

### 3. Explication de méthodes

#### • Cellule

Nous avons décidé d'associer chaque valeur (de 1 à 6) à son type et son degré dans la classe Cellule. Nous en avons profité pour régler l'état de toutes les cellules à false (la forêt initiale n'étant pas en feu).

#### • Program

##### Explication InitialiserFeu :

```
int positionL = 0;
int positionC = 0;
Random rand = new Random();
do
{
    positionL = rand.Next(0, surface.GetLength(0));
    positionC = rand.Next(0, surface.GetLength(1));

    } while (surface[positionL, positionC].Type == '/' ||
    surface[positionL, positionC].Type == '+');

    surface[positionL, positionC].Etat = true;
```

Le plus important ici est la boucle do while puisque sans elle la case aléatoire peut tomber sur de l'eau ou du terrain alors que ces derniers ne peuvent pas prendre feu donc il faut exclure ces 2 cas pour ne pas bloquer le programme.

##### Explication NbVoisinsFeu :

Nous avons commencé par parcourir la matrice "surface" pour stocker les coordonnées de la case qui a été mise en feu par la fonction InitialiserFeu. Nous avons donc attribué celles-ci respectivement dans indexligne et indexcolonne, qui sont deux paramètres d'entrée de la fonction.

```
for (int k = 0 ; k <= ligne; k++)
{
    for (int h = 0 ; h <= colonne; h++)
    {
        if (surface[k, h].Etat == false)
        {
            continue;
        }
        else if (surface[k, h].Etat == true)
        {
            indexligne = k;
            indexcolonne = h;
        }
    }
}
```

Pour déterminer les cases voisines de celle en feu nous avons ensuite déterminé qu'il fallait créer deux boucles for permettant d'entourer la case contaminée, c'est-à-dire créer un parcours tel que :

```
for (int i = indexligne - 1; i <= indexligne + 1 && i < ligne ; i++)  
for (int j = indexcolonne - 1; j <= indexcolonne + 1 && j <  
colonne; j ++)
```

Nous avons précédemment assigné indexligne et indexcolonne aux coordonnées de la case en feu. Nous utilisons donc ces deux variables pour parcourir l'entourage de la case choisie auparavant.

## 4. Contribution de chacun

Tout d'abord, nous avons débuté par une réunion brainstorming. Le but était de rassembler toutes les idées que l'on pouvait avoir et de sélectionner les pistes pertinentes. À la fin de cette réunion nous nous sommes divisé les tâches en fonction de nos compétences. Par ailleurs, il est important de noter que chacun a apporté son style dans le code final, et qu'il y a eu, outre la division des tâches, une réelle entraide quant à la compréhension des méthodes et du codage à réaliser.



## CONCLUSION

*Ce projet nous a principalement permis de nous rendre plus à l'aise vis-à-vis de la programmation. Toute notre équipe n'a jamais (ou peu) programmé, ce projet a donc été pour nous l'occasion de nous améliorer dans notre assimilation de cette nouvelle discipline.*

*Concernant les améliorations éventuelles de jeu, nous avons tous pensé tout d'abord à une interface graphique agréable pour le joueur. En se renseignant sur le sujet, nous avons vu qu'il était possible de réaliser cela grâce au langage XAML qui permet de construire des interfaces pour application mobile par exemple. Malheureusement la réalisation de cette interface a vite été mise de côté car nous devions d'abord avoir un code fonctionnel pour pouvoir penser aux améliorations.*