

Cahier des charges

Application de simulation de feux de forêts

Alexis REMY - Ewen HEAS - Tom ROBERT

B3 DEV WEB - MDS

Contexte

Dans le cadre de notre projet d'étude, notre objectif est de développer une application de simulation de feux de forêt basée sur des automates cellulaires pour aider les utilisateurs à prendre des décisions éclairées grâce aux résultats obtenus et grâce à l'interface graphique représentant une forêt et les feux.

Objectifs

Le principal objectif de cette application est de simuler la propagation des feux de forêt en prenant en compte différents paramètres tels que l'humidité du sol, le type de terrain et la force du vent.

L'application doit également permettre aux utilisateurs de visualiser dynamiquement l'évolution de la simulation, d'enregistrer les résultats, afin d'analyser les statistiques.

Acteurs

Un seul acteur : Groupe de développement (Alexis REMY, Ewen HEAS et Tom ROBERT)

Périmètre et Cas d'Utilisation

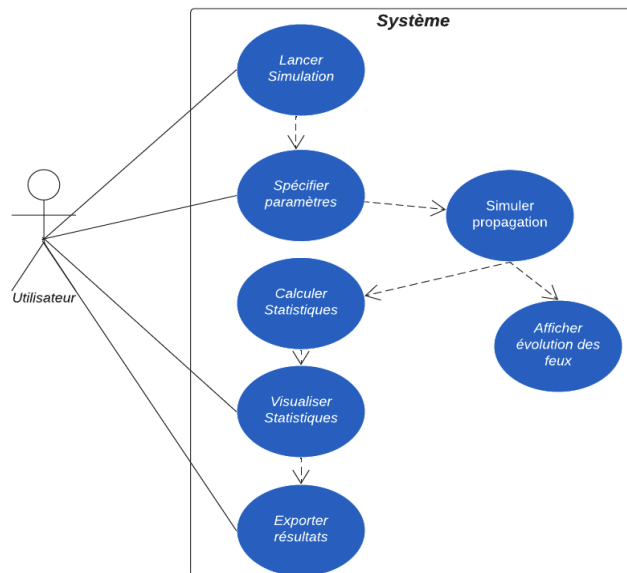
Cas d'Utilisation Principal :

L'utilisateur lance une simulation de feu de forêt en spécifiant les paramètres tels que la durée de la simulation, la force du vent, l'humidité, et le type de terrain initial.

L'application simule alors la propagation du feu en affichant dynamiquement l'évolution sur une grille. Une fois la simulation terminée, l'utilisateur peut visualiser les statistiques finales et exporter les résultats.

Diagramme cas utilisation - Feux de forêts

Alex R | May 2, 2024



Règles - Logique Métier

La probabilité d'enflammer une case dépend de l'humidité du sol et de la présence de végétation ou d'endroits vides.

Une case enflammée reste active pendant deux itérations, puis devient brûlée et chaude.

Une case brûlée et chaude peut envoyer des brandons aux cases voisines avec une probabilité dépendant de la force du vent.

Les cases brûlées et chaudes peuvent devenir brûlées et froides avec une certaine probabilité.

Exigences Fonctionnelles

L'application doit permettre à l'utilisateur de spécifier les paramètres de la simulation (durée, force du vent, humidité, type de terrain).

L'application doit simuler la propagation du feu sur une grille en fonction des paramètres spécifiés.

L'utilisateur doit pouvoir visualiser dynamiquement l'évolution de la simulation.

L'application doit fournir des statistiques finales sur l'état de la simulation.

L'utilisateur doit pouvoir exporter les résultats de la simulation.

L'application doit permettre de sauvegarder et de charger des simulations précédentes.

Exigences Non Fonctionnelles

L'interface utilisateur doit être intuitive et conviviale.

L'application doit être rapide et réactive même avec de grandes grilles de simulation.

L'exportation des résultats doit se faire de manière efficace et dans un format standard.

L'application doit être compatible avec les systèmes d'exploitation courants.

Contraintes

L'application doit être développée en Dart avec le framework Flutter.

Prestations Attendues

Livraison d'une application fonctionnelle répondant aux exigences spécifiées.

Documentation complète incluant le cahier des charges, le manuel utilisateur et la documentation technique.

Dictionnaire des Données

Case : Une unité de la grille de simulation. Correspond à l'objet ForestCell avec les attributs state, row, et col.

Humidité : Niveau d'humidité du sol, affectant la probabilité d'enflammage. Correspond à l'attribut humidity dans l'application.

Vent: Force et direction du vent, affectant la propagation du feu. Correspond aux attributs windStrength et windDirection dans l'application.

Grid: La grille de simulation, représentée par une liste de listes de ForestCell.

Documentation Technique

- ❖ Diagramme d'état/transition d'un feu de forêt: Décrivant les différents états possibles d'une case de la grille pendant la simulation :
 - ForestCellState.vide
 - ForestCellState.inflammable
 - ForestCellState.enFeu
 - ForestCellState.enFeuAvance
 - ForestCellState.brulee

- ForestCellState.eteint
- ForestCellState.inerte

Stack Technique

Langage de Programmation: Dart

Framework: Flutter

Détails Techniques

1. Classes et Objets :

- ❖ **ForestCell** : Représente une cellule de la grille.
 - Elle prend comme attributs :
 - ForestCellState state : État de la cellule.
 - int row : Position de la cellule (ligne).
 - int col : Position de la cellule (colonne).
- ❖ **ForestCellState** : Liste des valeurs représentant les différents états possibles d'une cellule :
 - vide
 - inflammable
 - enFeu
 - enFeuAvance
 - brulee
 - eteint
 - inerte

2. Paramètres de Simulation :

- ❖ **gridSize**: Taille de la grille de simulation.
- ❖ **windStrength**: Force du vent.
- ❖ **windDirection**: Direction du vent.
- ❖ **humidity**: Niveau d'humidité du sol.
- ❖ **emptyRatio**: Ratio de cellules vides.
- ❖ **maxIterations**: Nombre maximal d'itérations de la simulation.
- ❖ **currentIteration**: Compteur d'itérations en cours.
- ❖ **autoIterations**: Nombre d'itérations automatiques.

3. Fonctionnalités Principales :

- ❖ **initializeGrid():** Initialise la grille avec les états des cellules en fonction des paramètres définis.
- ❖ **startSimulation():** Démarre la simulation en réinitialisant les paramètres nécessaires.
- ❖ **simulateIteration():** Simule une itération de la propagation du feu.
- ❖ **spreadFire():** Gère la propagation du feu d'une cellule à ses voisines.
- ❖ **applyWindEffect():** Applique l'effet du vent sur la propagation du feu.
- ❖ **updateCell():** Met à jour l'état d'une cellule en fonction de son état actuel et de ses voisins.
- ❖ **deepCopyGrid():** Crée une copie profonde de la grille actuelle pour sauvegarder l'état de la simulation.
- ❖ **getNeighbors():** Récupère les cellules voisines d'une cellule donnée.
- ❖ **getNeighborsInDirection():** Récupère les cellules voisines dans une direction donnée.

4. Interface Utilisateur :

Grille de Simulation : Affichage de la grille représentant les différents états des cellules.

- ❖ **Contrôles de Simulation :**
 - Slider pour la taille de la grille.
 - Slider pour le type de terrain.
 - Slider pour la force du vent.
 - Dropdown pour la direction du vent.
 - Slider pour le niveau d'humidité.
 - Slider pour le nombre d'itérations automatiques.
- ❖ **Boutons :**
 - Lancer l'itération
 - Redémarrer la simulation
 - Revenir en arrière