

Simulation de propagation d'incendie

Sommaire

Objectif.....	1
Stockage des données	1
Description de l'interface.....	2
Affichage des données.....	2
Le programme principal	2
Modification des données	2

Objectif

Le but de ce projet est une simulation graphique de la propagation d'un feu. Du point de vue de la programmation le but est de renforcer l'habitude du découpage en fonction et de la programmation modulaire, ainsi que de parfaire la connaissance de la manipulation des tableaux au travers des types abstraits de données.

Stockage des données

On définira L et C comme constantes. On prendra $L=40$ et $C=60$.

Le terrain est divisé en $L \times C$ parcelles, chaque parcelle pouvant être de différents types :

- Terre
- Arbre
- Eau
- Feu
- Cendres
- Cendres éteintes

Au niveau de la représentation graphique, chaque parcelle sera représentée par une couleur correspondant au type de parcelle :

- Terre : Marron
- Arbre : Vert
- Eau : Bleu
- Feu : Rouge
- Cendres : Gris
- Cendres éteintes : Noir

- 1- Ecrire le type `t_parcelle` permettant de stocker une parcelle
- 2- Ecrire le type `t_terrain` permettant de stocker les $L \times C$ parcelles.

Il faut pouvoir remplir les parcelles et comme il y en a beaucoup, on va les remplir au hasard. Vous allez nourrir l'interface de deux fonctions, une qui remplit une parcelle au hasard, l'autre qui remplit le tableau. Et pour faire vivre le feu, il faudra également enrichir l'interface de nombreuses données et fonctionnalités.

Programmation modulaire

Vous organiserez votre application dans le respect de la programmation modulaire vue en cours.

Description de l'interface du module « terrain »

Dans l'hypothèse du développement d'un module terrain dans votre application, l'interface de ce module, comme nous l'avons vu en cours, contient les prototypes des fonctionnalités offertes aux codeurs qui l'exploitent.

On aura par exemple :

int couleur_aleatoire() qui renvoie une couleur au hasard dans l'ensemble des couleurs possibles du terrain, chaque couleur ayant la même probabilité d'apparaître.

void terrain_aleatoire() qui remplit aléatoirement toutes les parcelles du terrain.

Affichage des données

Maintenant que le terrain est rempli, on veut pouvoir l'afficher. Pour cela vous allez créer deux fonctions, une pour afficher une parcelle, l'autre pour afficher tout le terrain.

Vous complétez cette interface avec toutes les fonctionnalités qui vous sembleraient nécessaires (et vous les coderez dans la partie code local évidemment). Vous développerez également tout autre module susceptible d'améliorer l'organisation de votre application.

Programme principal

Donnez le programme principal qui permet d'initialiser le tableau et de l'afficher. Le programme principal organise les appels aux fonctionnalités de l'interface de chaque module afin de déclencher le feu et de suivre la propagation.

Modification des données

Le but étant de simuler la propagation d'un incendie, on va modifier les données. Le principe consiste à avoir deux tableaux T1 et T2. On initialise T1 puis on l'affiche. On calcule T2 à partir de T1, on recopie T2 dans T1, puis on affiche T1 et on recommence : calcul de T2 à partir de T1 , etc.

Il faut donc donner deux informations :

- 1- Les règles de calcul de T2 à partir de T1
- 2- Jusqu'à quand continue-t-on cette répétition de calcul et d'affichage ?

Le calcul de T2 se fait case par case. Pour chaque case de T1 de coordonnées (i, j) (ième ligne, jème colonne) on regarde les huit cases centrées autour de la case (i, j). Plusieurs cas sont possibles :

- 1- Si la parcelle (i, j) est en feu dans T1 elle devient en cendre dans T2.
- 2- Si la parcelle (i, j) est en cendres dans T1 elle devient en cendres éteintes dans T2.
- 3- Si la parcelle (i, j) est de type Arbre et qu'une des 8 parcelles autour est en feu, la parcelle devient en feu dans T1.
- 4- Une parcelle Eau reste toujours une parcelle Eau et une parcelle Terrain reste toujours une parcelle Terrain.

Extensions

En soutenance, après avoir montré une application qui respecte scrupuleusement le cahier des charges, vous êtes bien entendu libres d'améliorer le visuel de votre application en modifiant l'aspect des parcelles décrites ci-dessus, dans le respect de l'esprit du sujet.

Contraintes matérielles et logicielles

Méthode de conduite de projet :

- Modèle / Vue / Contrôleur

Outil de planification exigé :

- MS-project (licence ELMS disponible) ou équivalent

Langage imposé : C et Allegro pour l'aspect graphique.

Matériel :

- Vos PC personnels ou ceux de l'école. La soutenance se fera sur la machine de votre choix (testez-la avant !).

Planning et organisation du travail

Période de réalisation du projet : début octobre jusqu'à la semaine du 2 décembre.

Équipes : en trinôme ou éventuellement en binôme.

Évaluation : évaluation en continu (dans le cadre des « notes de suivi ») du travail et de l'organisation de l'équipe, de l'avancement et de la qualité de l'analyse et de la production.

Ce projet donnera lieu à une soutenance : 15 minutes de présentation par équipe sur vidéoprojecteur. Nous vous précisons les modalités de cette soutenance.

Un compte-rendu de 20 pages maximum reprenant les grandes étapes d'analyse ci-dessous doit être imprimé le jour de la soutenance. Pour plus de détails, voir la dernière page.