

Project Autonome : Project B Incendie

Matthieu Barnabé

Luc Maillard



Sommaire

- I. Présentation
- II. Cahier des charges
- III. Devis
- IV. Explication du code

I. Présentation

Le projet B a pour but de simuler la progression d'incendie et de voir comment se comporte un incendie. En terme technique, ce projet sert à améliorer l'utilisation de tableaux

II. Cahier des charges

En plus de la consigne de l'énoncé, nous avons rajouter un mode qui permet de simuler une propagation plus logique d'un incendie : c'est-à-dire que le terrain est composé seulement de terre, arbre et eau puis nous on choisit une case à mettre en feu et on applique les règles de l'énoncé.

en plus de ce mode, nous avons rajouter un calcul de moyenne sur plusieurs simulations, les données qui nous intéresse sont le pourcentage d'arbre restant par rapport au nombre de départ et le nombre de boucle pour que le feu s'éteigne. nous avons bien appliqué cette simulation aux deux règles de simulation (celle de l'énoncé et celle dite réel) Seule la première simulation est actuellement fonctionnelle.

Ce qui reste à développer est la seconde simulation et une animation de la transmission du feu entre chaque case.

III. Devis

Matthieu
3 rue Daru
75008 Paris
Siret :
N° TVA :
Tél :

| | |
|---------------|------------|
| Date du devis | 18/01/2021 |
|---------------|------------|

| Description | Quantité | Prix unitaire HT | Prix total HT |
|---|----------|------------------|---------------|
| Développement et simulation d'un incendie Développement d'une application permettant de simuler un incendie | 1 | 80,00 € | 80,00 € |
| Affichage graphique | 1 | 50,00 € | 50,00 € |

| | |
|------------------|-----------------|
| Total HT | 130,00 € |
| TVA (20,00 %) | 26,00 € |
| Total TTC | 156,00 € |

IV. Explication du code

fichier main.c :

```
menu_start(RT,origin);
srand(time(NULL));
int choix_menu = switch_menu();
//génération du terrain en fonction du mode
if (choix_menu == 1) terrain_aleatoire(&terrain);
if (choix_menu == 2) mode2_generation(&terrain);
//affichage du terrain et initialisation du tableau de stats
if (choix_menu == 1 || choix_menu == 2)
{
    printf("\n      Terre      |  Arbre      |      Eau      |      Feu      |  Cendres
    affichage_nb_case(terrain,rep2);
    AffichageAllegro(terrain,RT,origin);
}
//choix de la case a mettre en feu
if (choix_menu == 2) mode2_clic(&terrain);
//modifications des données
if (choix_menu == 1 || choix_menu == 2)
{
    while (rep == 1 )
    {
        rep2++;
        if (nb_feu == 0) rep = 0;
        nb_feu = gestion_feux(terrain,&terrain2);
        gestion_t1(&terrain,terrain2);
        Sleep(1000);
        AffichageAllegro(terrain,RT,origin);
        affichage_nb_case(terrain,rep2);
    }
}
```

le
main

permet de créer la fenetre Allegro et de gerer les différents mode et la gestion du feu

fichier parcelles.c : génération du terrain et affichage sur fenetre Allegro

```
//permet de renvoyer une couleur au hasard
char couleur_aleatoire()
{
    char couleur[20][6] = {"Terre", "Arbre", "Eau", "Feu", "Cendres", "Cendres éteintes"};
    srand(time(NULL));
    int x = rand() % 6;
    printf("\n%d", x);
    printf("\n%s", couleur[x]);
    return couleur[x];
}
//fonction de generer le terrain
void terrain_aleatoire(t_terrain *terrain)
{
    int i,j;
    for (i=0;i<40;i++)
    {
        for (j=0;j<60;j++)
        {
            terrain->terrain[i][j] = rand() % 6;
        }
    }
}
//affichage du terrain dans allegro
void AffichageAllegro(t_terrain terrain,int RT,int origin)
{
    int i,j;
    for (i = 0; i<40; i++)
    {
        for (j=0;j<60;j++)
        {
            switch(terrain.terrain[i][j])
            {
                case 0 :
                    rectfill(screen,RT*(j+origin),RT*i,RT*(j+1+origin),RT*(i+1),makecol(106,61,51)); //terre
                    break;
                case 1 :
                    rectfill(screen,RT*(j+origin),RT*i,RT*(j+1+origin),RT*(i+1),makecol(81,116,63)); //arbre
                    break;
                case 2 :
                    rectfill(screen,RT*(j+origin),RT*i,RT*(j+1+origin),RT*(i+1),makecol(40,67,141)); //eau
                    break;
                case 3 :
                    rectfill(screen,RT*(j+origin),RT*i,RT*(j+1+origin),RT*(i+1),makecol(255,0,0)); //feu
                    break;
                case 4 :
                    rectfill(screen,RT*(j+origin),RT*i,RT*(j+1+origin),RT*(i+1),makecol(159,161,165)); //cendres
            }
        }
    }
}
```

fichier parcelles.c : gestion du feu et modif du tableau t1

```
//gestion la progression du feu
int gestion_feu(t_terrain terrain1, t_terrain *terrain2)
{
    int i,j;
    int xi[8] = {-1,-1,-1,0,0,1,1,1};
    int yi[8] = {-1,0,1,-1,1,-1,0,1};
    int nb_feu = 0;
    for (i = 0;i<40;i++)
    {
        for (j=0;j<60;j++)
        {
            terrain2->terrain[i][j] = terrain1.terrain[i][j];
            if (terrain1.terrain[i][j] == 1){
                for (int k=0;k<8;k++){
                    if (terrain1.terrain[i-xi[k]][j-yi[k]] == 3) terrain2->terrain[i][j] = 3;}}
            if (terrain1.terrain[i][j] == 3) terrain2->terrain[i][j] = 4;
            if (terrain1.terrain[i][j] == 4) terrain2->terrain[i][j] = 5;
            if (terrain2->terrain[i][j]== 3) nb_feu++;
        }
    }
    return nb_feu;
}

//gestion de modifier le terrain 1 une fois que les regle de gestion du feu ont été appliqué sur le terrain 2
void gestion_t1(t_terrain *terrain1,t_terrain terrain2)
{
    int i,j;
    for (i=0;i<40;i++)
    {
        for(j=0;j<60;j++)
        {
            terrain1->terrain[i][j] = terrain2.terrain[i][j];
        }
    }
}
```

fichier parcelles.c : affichage du nombre de case par type

```
//affichage sur le nombre de case par type
void affichage_nb_case(t_terrain terrain,int rep)
{
    int nb_terre = 0;
    int nb_arbre = 0;
    int nb_eau = 0;
    int nb_feu = 0;
    int nb_cendres = 0;
    int nb_cendres_eteintes = 0;
    int i,j;
    for (i = 0;i<40;i++){
        for (j=0;j<60;j++){
            switch(terrain.terrain[i][j]){
                case 0 :
                    nb_terre++;
                    break;
                case 1 :
                    nb_arbre++;
                    break;
                case 2 :
                    nb_eau++;
                    break;
                case 3 :
                    nb_feu++;
                    break;
                case 4 :
                    nb_cendres++;
                    break;
                case 5 :
                    nb_cendres_eteintes++;
                    break;}}}
    char a;
    a = '\n';
    printf("\n%d %d %5.2f%c | %d %5.2f%c | %d %5.2f%c ",rep,nb_terre, nb_terre/24.0,a,nb_arbre, nb_arbre/24.0,a,nb_eau,nb_ea
    printf("| %d %5.2f%c | %d %5.2f%c | %d %5.2f%c",nb_feu,nb_feu/24.0,a,nb_cendres,nb_cendres/24.0,a,nb_cendres_eteintes,nb_c
    int origin = 15*3;
    fond_menu(14,10);
    textprintf_ex(screen,font,10,origin,makecol(255,255,255),-1,"Terre : %-3d",nb_terre);
    textprintf_ex(screen,font,74,origin+10,makecol(255,255,255),-1,"%-5.2f %c",nb_terre/24.0,a);

    textprintf_ex(screen,font,10,origin+50,makecol(255,255,255),-1,"Arbre : %-3d",nb_arbre);
    textprintf_ex(screen,font,74,origin+60,makecol(255,255,255),-1,"%-5.2f %c",nb_arbre/24.0,a);
```