Algorithmique et Programmation

—

Examen sur machine: solution

—

G1: monasse*(at)*certis.enpc.fr G2: thorsten*(at)*certis.enpc.fr
G3: aganj*(at)*certis.enpc.fr G4: tousch*(at)*certis.enpc.fr
G5: courchay*(at)*certis.enpc.fr G6: eric.bughin*(at)*gmail.com

—

12/12/08

—

```
1  // ================ EXAMEN MACHINE 2008/2009 ==============
2
3  #include <CL/Graphics/Graphics.h>
4  #include <iostream>
5
6  using namespace CL::Graphics;
7  using namespace std;
8
9  // ================= COMPLEXES =================
10 // Structure
11 struct complexe {
12     double a,b;
13 };
14 // Addition
15 complexe plus(complexe z1,complexe z2) {
16     complexe z={z1.a+z2.a,z1.b+z2.b};
17     return z;
18 }
19 // Multiplication
20 complexe fois(complexe z1,complexe z2) {
21     complexe z={z1.a*z2.a-z1.b*z2.b,z1.a*z2.b+z1.b*z2.a};
22     return z;
23 }
24 // Module
25 double module(complexe z) {
26     return sqrt(z.a*z.a+z.b*z.b);
27 }
28 // ================ Mandelbrot ==================
29 Color Mandelbrot(int i,int j,int n) { // Pixel (i,j) dans [0,n]^2
30     int max_iter=20;                  // Nbre max d'iteration avant de déclarer
31                                       // la non divergence
32     double x0=-2,x1=1,y0=-1.5,y1=1.5;
33     complexe c={(x1-x0)*i/n+x0,(y1-y0)*j/n+y0}; // c dans [x0,x1]+i[y0,y1]
34     complexe z={0,0};                           // z(0)
35     int it=0;
36     while (module(z)<=2 && it <max_iter) {// divergence si module >2
37         z=plus(fois(z,z),c);              // z(n+1)=z(n)^2+c
38         it++;
39     }
40     byte grey=byte(it*255/max_iter);      // couleur de noir a blanc en
41                                           // fonction du nbre d'iterations
```

```
42      return Color(grey,grey,grey);
43  }
44  // =============== Dessins ======================
45  void dessin_direct(int n)
46  {
47      for (int i=0;i<n;i++)             // Direct pour tous les pixels
48          for (int j=0;j<n;j++)
49              DrawPoint(i,j,Mandelbrot(i,j,n));
50
51  }
52
53  void dessin_progressif(int n)         // Affichage progressif
54  {
55      for (int s=n;s>=1;s/=2)           // tous les s pixels
56          for (int i=0;i<n;i+=s)
57              for (int j=0;j<n;j+=s)
58                  if (s==n || (i%(2*s))!=0 || (j%(2*s))!=0)// si pas deja affiche
59
60                      FillRect(i,j,s,s,Mandelbrot(i,j,n)); // rectangle de la couleur
61                                                           // calculee pour (i,j)
62  }
63  // ============== Main ============================
64  int main()
65  {
66      OpenWindow(512,512);
67      dessin_direct(256);
68      Click();
69      Clear();
70      dessin_progressif(512);
71      Terminate();
72      return 0;
73  }
```