

Algorithmique et Programmation

Examen sur machine - Solution

G1: keriven(at)certis.enpc.fr G2: juan(at)certis.enpc.fr
G3: gmellier(at)melix.org G4: pierre.maurel(at)ens.fr
G5: pierre(at)senellart.com G6: adde(at)certis.enpc.fr

07/01/05

1 Solution

Fichier vect.h

```
1 #pragma once
2
3 struct vect {
4     double x,y,z;
5 };
6
7 vect operator+(vect a, vect b);
8 vect rotate(vect a, double alpha);
```

Fichier vect.cpp

```
1 #include "vect.h"
2 #include <cmath>
3
4 vect operator+(vect a, vect b) {
5     vect c={a.x+b.x,a.y+b.y,a.z+b.z};
6     return c;
7 }
8
9 vect rotate(vect a, double alpha) {
10     vect c={
11         a.x*cos(alpha)-a.y*sin(alpha),
12         a.x*sin(alpha)+a.y*cos(alpha),
13         a.z
14     };
15     return c;
16 }
```

Fichier tetra.h

```
1 #pragma once
2
3 #include "vect.h"
4 #include "camera.h"
5
6 #include <win>                    // Pour la couleur
7 using namespace Win;
8
```

```

9  struct tetra {
10     Color c;
11     vect M[4];
12 };
13
14 void dessine(camera C,tetra T,Color col);
15 void affiche(camera C,tetra T);
16 void efface(camera C,tetra T);
17
18 tetra regulier();
19 tetra translate(tetra T,vect t);
20 tetra rotate(tetra T,double alpha);
21 tetra changeColor(tetra T,Color c);

```

Fichier tetra.cpp

```

1  #include "tetra.h"
2
3  void dessine(camera C,tetra T,Color col) {
4      Pixel m[4];
5      for (int i=0;i<4;i++)
6          m[i]=projette(C,T.M[i]);
7      for (int i=0;i<3;i++)
8          for (int j=i+1;j<4;j++)
9              DrawLine(m[i],m[j],col);
10 }
11
12 void affiche(camera C,tetra T) {
13     dessine(C,T,T.c);
14 }
15
16 void efface(camera C,tetra T) {
17     dessine(C,T,White);
18 }
19
20 tetra regulier() {
21     tetra T={
22         Red,
23         {
24             {1,0,0},
25             {-0.5,sqrt(3.)/2,0},
26             {-0.5,-sqrt(3.)/2,0},
27             {0,0,sqrt(2.)}
28         }
29     };
30     return T;
31 }
32
33 tetra translate(tetra T,vect t) {
34     tetra T2;
35     T2.c=T.c;
36     for (int i=0;i<4;i++)
37         T2.M[i]=T.M[i]+t;
38     return T2;
39 }
40
41
42 tetra rotate(tetra T,double alpha) {
43     tetra T2;
44     T2.c=T.c;
45     for (int i=0;i<4;i++)
46         T2.M[i]=rotate(T.M[i],alpha);

```

```

47     return T2;
48 }
49
50 tetra changeColor(tetra T,Color c) {
51     tetra T2=T;
52     T2.c=c;
53     return T2;
54 }

```

Fichier camera.h

```

1  #pragma once
2
3  #include "vect.h"
4  #include <win>           // Pour les Pixels
5  using namespace Win;
6
7  struct camera {
8      int u0,v0;         // centre de l'image
9      double x0;        // eloignement
10     double f;         // focale
11 };
12
13 Pixel projette(camera c,vect M);

```

Fichier camera.cpp

```

1  #include "camera.h"
2
3  Pixel projette(camera c,vect M) {
4      Pixel m;
5      m.x=int(c.u0+c.f*M.y/(c.x0-M.x));
6      m.y=int(c.v0-c.f*M.z/(c.x0-M.x));
7      return m;
8  }

```

Fichier main.cpp

```

1  #include <win>
2  using namespace Win;
3
4  #include "vect.h"
5  #include "tetra.h"
6  #include "camera.h"
7  int main()
8  {
9      vect trans[4]={
10         {1,0,0},
11         {-0.5,sqrt(3.)/2,0},
12         {-0.5,-sqrt(3.)/2,0},
13         {0,0,sqrt(2.)}
14     };
15     Color col[4]={Red,Blue,Green,Black};
16     tetra tet[4];
17     for (int j=0;j<4;j++)
18         tet[j]=changeColor(translate(regulier()),trans[j],col[j]);
19
20     camera C={256,256,10,500};
21     OpenWindow(512,512);
22
23     for (int i=0;i<10000;i++) {
24         vect t={-12+8*cos(i/150.),8*sin(i/150.0),-3.0};
25         NoRefreshBegin();

```

```
26         for (int j=0;j<4;j++)
27             affiche(C,translate(rotate(tet[j],i/15.0),t));
28         NoRefreshEnd();
29         MilliSleep(10);
30         NoRefreshBegin();
31         for (int j=0;j<4;j++)
32             efface(C,translate(rotate(tet[j],i/15.0),t));
33         NoRefreshEnd();
34     }
35     Terminate();
36     return 0;
37 }
```