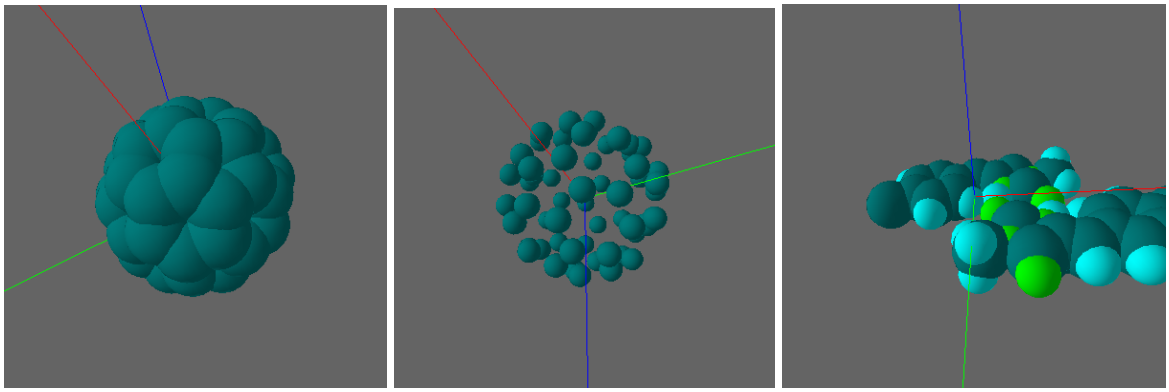
CNAM NSY116 - Multimédia et interaction humain-machine

ED 7 - 3D interactive



Exercice 1 : Fonctions de base 3D avec Processing

1) Ecrire un programme minimal qui affiche un empilement de spheres.

2) Modifier le programme pour qu'il produise une scène 3D à partir d'informations issues d'un fichier au format .xyz. Ce format standard décrit les noms et les positions des atomes qui composent des molécules. On en trouve de nombreux exemples sur internet (avec plusieurs visualiseurs)

```
C     5.13090  -3.00610  -0.57175
H     1.34755  -0.49373  -0.57175
H    -0.01641  -4.18763  -0.57175
H    -2.26947  -4.20572  -0.57175
H    -4.48901  -3.09882  -0.57175
Cl   -5.15734  -0.35876  -0.57176
H    -2.57167   0.76111  -0.57176
H    -0.36495  -0.30581  -0.57176
O     2.67768  -4.74775  -0.57175
Se    4.29633   0.14195  -0.57175
H     5.74708  -2.08996  -0.57176
```

3) Comment faire pour être assuré que la scène sera entièrement visible à l'écran ?

Exercice 2 : Navigation dans la scène

1) Décrire différentes manières de naviguer dans la scène 3D

2) Modifier le programme pour permettre une interaction en mode "study"

3) Au-delà des modes d'interaction, comment procurer à l'utilisateur une meilleure compréhension de la scène 3D ?

1.1)

The default values are camera(width/2.0, height/2.0, (height/2.0) / tan(PI*60.0 / 360.0), width/2.0, height/2.0, 0, 0, 1, 0).

1.2) On peut utiliser un petit script Perl qui lit le fichier de molecule et produit le code processing correspondant à la structure de donnée qui stocke la scène (ici, juste les positions + rayons + couleurs des sphères). On colle ensuite ce bout de code dans le sketch processing. Dans un "vrai" programme, on ferait la lecture du fichier .xyz en interne.

```
#!/usr/bin/perl
## xyz2PDE
## un convertisseur pour dessiner des structures moleculaires
## avec processing
$n=0;

print "float [][] atomes = { \n";
while (<>) {
 # chaque atome est traduit par une sphere
 ($nom,$x,$y,$z) = split;
 # la couleur depend du type d'atome
 @cols=(255,0,0);
 if (substr($nom,0,1) eq "H") {@cols=(255,255,255);} #blanc
 if (substr($nom,0,1) eq "C") {@cols=(127,127,127);} #gris
 if (substr($nom,0,2) eq "SI") {@cols=(255,200,0);}  #orange
 # le rayon aussi, mais on verra plus tard les bonnes valeurs
 $r = 1.0;
 if (substr($nom,0,1) eq "H") {$r = 1.0;}
 if (substr($nom,0,1) eq "C") {$r = 1.5;}
 if (substr($nom,0,2) eq "SI") {$r = 2.0;}
 # production de la ligne PDE
 print "{$x,$y,$z,$r,$cols[0],$cols[1],$cols[2]}, // atome $nom\n";
 $n++;
}
print "}; \n";
print "int nbAtomes = $n; \n";
```

à l'exécution, on obtient :

```
pcubaud$ perl xyz2pde.pl <C2h2.xyz
float [][] atomes = {
{1.37563,-2.65510,-0.57176,1.5,127,127,127}, // atome C
{2.57940,-3.54162,-0.57175,1.5,127,127,127}, // atome C
{3.71340,-2.63261,-0.57176,1,255,0,0}, // atome N
{3.28260,-1.30688,-0.57176,1.5,127,127,127}, // atome C
{1.90426,-1.31897,-0.57176,1,255,0,0}, // atome N
```

etc.

1.3) une solution pour poser la caméra à un bon endroit dans la scène, consiste à viser le centre de gravité de la scène et à reculer la caméra derrière le volume englobant de la scène.

Le code Processing du visualiseur :

```
import processing.opengl.*;


float xmax,ymax,zmax,xgrav,ygrav,zgrav,sx,sy,sz;

void setup(){
  size(400,400,OPENGL);
  sphereDetail(3);
  frameRate(20);

  // calcul barycentre
  sx = sy = sz = 0;
  for (int i=0;i<nbAtomes;i++){
    sx += atomes[i][0]; sy += atomes[i][1]; sz += atomes[i][2];
  }
  xgrav = sx/nbAtomes;ygrav = sy/nbAtomes;zgrav = sz/nbAtomes;
  // recherche englobant
  xmax = ymax = zmax = -1000;
  for (int i=0;i<nbAtomes;i++){
    xmax = max(xmax, atomes[i][0]+atomes[i][3]);
    ymax = max(ymax, atomes[i][1]+atomes[i][3]);
    zmax = max(zmax, atomes[i][2]+atomes[i][3]);
  }
}

void draw(){
  lights();
  background(200);
  camera(xmax+5,ymax+5,zmax+5,xgrav,ygrav,zgrav,0.0,1.0,0.0);
  dessineRepere();
  pushMatrix();
  translate(xgrav,ygrav,zgrav);
  rotateY(map(mouseX,0,width,-PI,PI));
  rotateX(map(mouseY,0,height,-PI,PI));
  dessineRepere();
  dessineScene();
  popMatrix();
}

void dessineScene(){
  noStroke();
  for (int i=0;i<nbAtomes;i++){
    pushMatrix();
    translate(atomes[i][0],atomes[i][1],atomes[i][2]);
    fill(atomes[i][3],atomes[i][4],atomes[i][5]);
    sphere(atomes[i][3]);
    popMatrix();
  }
}

void dessineRepere(){
  stroke(255,0,0);
  line(0,0,0,200,0,0);
  stroke(0,255,0);
  line(0,0,0,0,200,0);
  stroke(0,0,255);
  line(0,0,0,0,0,200);
}

void keyPressed(){
```

```
  saveFrame("mol####.png");
}
```

Avec par exemple, pour le tableau atome[] :

```
float [][] atomes = {
{4.20600,5.95200,1.01500,2,255,200,0}, // atome SI1
{4.32400,3.68600,1.57700,2,255,200,0}, // atome SI2
{5.97700,2.69400,0.22600,2,255,200,0}, // atome SI3
{8.05600,3.61500,0.85700,2,255,200,0}, // atome SI4
{7.94900,5.86700,0.23200,2,255,200,0}, // atome SI5
{6.27300,6.97700,1.43900,2,255,200,0}, // atome SI6
{2.52500,7.05600,2.20700,2,255,200,0}, // atome SI7
{2.45000,9.28000,1.43800,2,255,200,0}, // atome SI8
{4.47300,10.32100,2.06100,2,255,200,0}, // atome SI9
{6.14800,9.24100,0.83600,2,255,200,0}, // atome SI0
{8.22800,10.25600,1.13700,2,255,200,0}, // atome SI1
{9.84400,9.17000,-0.18000,2,255,200,0}, // atome SI2
{9.99900,6.94000,0.54300,2,255,200,0}, // atome SI3
{2.29800,9.32000,-0.43400,1.5,127,127,127}, // atome C14
{1.73700,8.25300,-1.13900,1.5,127,127,127}, // atome C15
{1.64500,8.27800,-2.53000,1.5,127,127,127}, // atome C16
{2.11100,9.38100,-3.24500,1.5,127,127,127}, // atome C17
{2.66400,10.45700,-2.55200,1.5,127,127,127}, // atome C18
{2.75300,10.42400,-1.16100,1.5,127,127,127}, // atome C19
{9.34800,9.24000,-1.98700,1.5,127,127,127}, // atome C24
{8.14800,8.66700,-2.40800,1.5,127,127,127}, // atome C25
{7.75100,8.70800,-3.74200,1.5,127,127,127}, // atome C26
{8.56200,9.33200,-4.68900,1.5,127,127,127}, // atome C27
{9.76600,9.90900,-4.28600,1.5,127,127,127}, // atome C28
{10.15300,9.86200,-2.94500,1.5,127,127,127}, // atome C29
{5.67500,3.09000,-1.58500,1.5,127,127,127}, // atome C34
{4.39900,3.40600,-2.05900,1.5,127,127,127}, // atome C35
{4.18700,3.72200,-3.40000,1.5,127,127,127}, // atome C36
{5.25300,3.72300,-4.30000,1.5,127,127,127}, // atome C37
{6.53000,3.40000,-3.84200,1.5,127,127,127}, // atome C38
{6.73300,3.08500,-2.49900,1.5,127,127,127}, // atome C39
{6.76400,6.82600,3.74100,2,255,200,0}, // atome SI4
{2.93800,7.04200,4.49500,2,255,200,0}, // atome SI5
{4.80100,10.14800,4.35100,2,255,200,0}, // atome SI6
{5.07100,7.90100,4.94700,2,255,200,0}, // atome SI7
{8.87700,10.10000,3.37200,2,255,200,0}, // atome SI8
{10.56200,6.93300,2.80700,2,255,200,0}, // atome SI9
{8.42200,3.34300,3.13100,2,255,200,0}, // atome SI0
{4.87000,3.40000,3.82000,2,255,200,0}, // atome SI1
{6.83500,4.56600,4.34300,2,255,200,0}, // atome SI2
{8.83100,7.86600,4.08600,2,255,200,0}, // atome SI3
{5.04800,4.03300,-5.56200,1.5,127,127,127}, // atome C81
{4.83100,4.34000,-6.72400,1.5,127,127,127}, // atome C82
{2.03200,9.39500,-4.55800,1.5,127,127,127}, // atome C83
{1.96500,9.37700,-5.77700,1.5,127,127,127}, // atome C84
{8.15700,9.35800,-5.94000,1.5,127,127,127}, // atome C85
{7.71400,9.34600,-7.07800,1.5,127,127,127}, // atome C86
{4.56300,4.69500,-7.96200,1.5,127,127,127}, // atome C87
{3.23900,4.70000,-8.40300,1.5,127,127,127}, // atome C88
{2.93400,5.11100,-9.70000,1.5,127,127,127}, // atome C89
{3.94900,5.52100,-10.56600,1.5,127,127,127}, // atome C90
{5.27500,5.49800,-10.13100,1.5,127,127,127}, // atome C91
{5.58000,5.09000,-8.83200,1.5,127,127,127}, // atome C92
{7.17500,9.29800,-8.27600,1.5,127,127,127}, // atome C97
{7.83400,9.82200,-9.38900,1.5,127,127,127}, // atome C98
{7.23900,9.75800,-10.64900,1.5,127,127,127}, // atome C99
{5.98300,9.17000,-10.80400,1.5,127,127,127}, // atome C01
{5.32900,8.64900,-9.68900,1.5,127,127,127}, // atome C02
{5.91900,8.71100,-8.42800,1.5,127,127,127}, // atome C03
{1.90300,9.31900,-7.09000,1.5,127,127,127}, // atome C08
{1.25100,8.24700,-7.70200,1.5,127,127,127}, // atome C09
{1.21300,8.15400,-9.09200,1.5,127,127,127}, // atome C10
{1.82700,9.12800,-9.87900,1.5,127,127,127}, // atome C11
{2.46500,10.20900,-9.26900,1.5,127,127,127}, // atome C12
{2.50300,10.30300,-7.87700,1.5,127,127,127}, // atome C13
{3.63400,5.97700,-11.75900,1.5,127,127,127}, // atome C18
{3.29300,6.46700,-12.82600,1.5,127,127,127}, // atome C19
{1.85200,8.97900,-11.18700,1.5,127,127,127}, // atome C20
{1.92700,8.76300,-12.38600,1.5,127,127,127}, // atome C21
{5.41900,9.10200,-11.99000,1.5,127,127,127}, // atome C22
{4.90600,9.03900,-13.09700,1.5,127,127,127}, // atome C23
```

```
{4.33200,8.97800,-14.44400,1.5,127,127,127}, // atome C24
{2.04900,8.42300,-13.81200,1.5,127,127,127}, // atome C26
{2.82500,7.11000,-14.06400,1.5,127,127,127}, // atome C28
{3.99000,7.57300,-14.96400,1.5,127,127,127}, // atome C30
{2.88100,9.46500,-14.58900,1.5,127,127,127}, // atome C31
{3.41500,7.91500,-16.35900,1.5,127,127,127}, // atome C34
{2.65000,9.23000,-16.09900,1.5,127,127,127}, // atome C35
{1.35900,7.36500,-0.60500,1,255,255,255}, // atome H20
{1.20300,7.41600,-3.05900,1,255,255,255}, // atome H21
{3.03800,11.34100,-3.09900,1,255,255,255}, // atome H22
{3.19700,11.29000,-0.64200,1,255,255,255}, // atome H23
{7.48400,8.17000,-1.68800,1,255,255,255}, // atome H30
{6.79400,8.24400,-4.03700,1,255,255,255}, // atome H31
{10.41700,10.40800,-5.02400,1,255,255,255}, // atome H32
{11.10800,10.32700,-2.64900,1,255,255,255}, // atome H33
{3.53500,3.41500,-1.37500,1,255,255,255}, // atome H40
{3.16800,3.97400,-3.74200,1,255,255,255}, // atome H41
{7.38800,3.39300,-4.53700,1,255,255,255}, // atome H42
{7.75500,2.83500,-2.16700,1,255,255,255}, // atome H43
{3.91700,6.06900,-0.44100,1,255,255,255}, // atome H54
{3.01500,3.03900,1.29000,1,255,255,255}, // atome H55
{9.13400,2.94100,0.08300,1,255,255,255}, // atome H56
{7.59600,5.90300,-1.21300,1,255,255,255}, // atome H57
{1.21600,6.40500,1.92600,1,255,255,255}, // atome H58
{4.41800,11.75300,1.66200,1,255,255,255}, // atome H59
{5.76800,9.32800,-0.59900,1,255,255,255}, // atome H60
{8.15200,11.68600,0.73200,1,255,255,255}, // atome H61
{11.05400,6.24300,-0.24200,1,255,255,255}, // atome H62
{1.91800,7.87100,5.19500,1,255,255,255}, // atome H63
{2.77800,5.68400,5.07300,1,255,255,255}, // atome H64
{5.95500,10.96400,4.80800,1,255,255,255}, // atome H65
{3.63000,10.71300,5.07500,1,255,255,255}, // atome H66
{5.35000,7.80100,6.40500,1,255,255,255}, // atome H67
{10.25600,10.63700,3.53000,1,255,255,255}, // atome H68
{8.03400,10.95800,4.24200,1,255,255,255}, // atome H69
{10.89700,5.56500,3.27800,1,255,255,255}, // atome H70
{11.80700,7.72200,3.01600,1,255,255,255}, // atome H71
{8.31600,1.90400,3.49300,1,255,255,255}, // atome H72
{9.80200,3.73000,3.52000,1,255,255,255}, // atome H73
{3.75500,3.80900,4.71000,1,255,255,255}, // atome H74
{5.06900,1.95300,4.10900,1,255,255,255}, // atome H75
{7.11600,4.44200,5.79900,1,255,255,255}, // atome H76
{9.19000,7.80000,5.52800,1,255,255,255}, // atome H77
{5.99600,1.22300,0.45300,1,255,255,255}, // atome H78
{1.30900,9.99100,2.07600,1,255,255,255}, // atome H79
{11.16000,9.83600,0.02300,1,255,255,255}, // atome H80
{2.42200,4.39400,-7.72700,1,255,255,255}, // atome H93
{1.88000,5.12600,-10.02800,1,255,255,255}, // atome H94
{6.08700,5.82900,-10.80100,1,255,255,255}, // atome H95
{6.63100,5.09700,-8.49500,1,255,255,255}, // atome H96
{8.82900,10.28800,-9.27900,1,255,255,255}, // atome H04
{7.77200,10.17400,-11.52100,1,255,255,255}, // atome H05
{4.34000,8.17600,-9.79100,1,255,255,255}, // atome H06
{5.38300,8.28900,-7.56000,1,255,255,255}, // atome H07
{0.77500,7.45900,-7.09300,1,255,255,255}, // atome H14
{0.70900,7.29100,-9.56100,1,255,255,255}, // atome H15
{2.96500,10.98000,-9.88000,1,255,255,255}, // atome H16
{3.02700,11.15100,-7.40600,1,255,255,255}, // atome H17
{5.03400,9.52400,-15.11900,1,255,255,255}, // atome H25
{1.00800,8.36700,-14.20900,1,255,255,255}, // atome H27
{2.19300,6.35400,-14.58600,1,255,255,255}, // atome H29
{4.84500,6.85500,-15.00900,1,255,255,255}, // atome H32
{2.69500,10.52200,-14.28000,1,255,255,255}, // atome H33
{4.24200,8.07600,-17.09200,1,255,255,255}, // atome H36
{2.76000,7.10500,-16.75700,1,255,255,255}, // atome H37
{3.08800,10.06700,-16.69300,1,255,255,255}, // atome H38
{1.56700,9.15900,-16.35900,1,255,255,255}, // atome H39
};
int nbAtomes = 138;
```