```
// INFOGRAPHIE cours N° 5 // 21 novembre 2004
/*
 ESCALIER MOBILE  © alain marty 2004
 une trémie de 90/240,
 trois paliers avec portes vitrées à -120, 000, +120,
 8 marches mobiles coulissant sur 8 guides suivant 2 configurations :
 1) en volées d'escalier reliant les niveaux -120 à 000 et 000 à +120
 2) en plateau horizontal reliant les niveaux -120, 000 et +120
 chaque marche comprend un moteur électrique
 sa position est contrôlée électroniquement
 la sécurité est assurée par des portes vitrées asservies
*/

// 1) BIBLIOTHEQUE :

#macro axes()
 cylinder { <-10,0,0> <10,0,0> 0.01 pigment { color rgb <1,0,0> } }
 cylinder { <0,-10,0> <0,10,0> 0.01 pigment { color rgb <0,1,0> } }
 cylinder { <0,0,-10> <0,0,10> 0.01 pigment { color rgb <0,0,1> } }
#end

#macro mon_finish( reflec )
 finish
 { ambient 0.3
  diffuse 0.8
  specular 0.7
  roughness 0.01
  reflection reflec
 }
#end

#macro pigment_marbre()
 pigment
  {  granite // marble wood granite
   color_map
   { [ 0.4 color rgbt <0.7,0.4,0.0> ]
    [ 0.5 color rgbt <0.2,0.1,0.2> ]
    [ 0.6 color rgbt <0.2,0.1,0.2> ]
   }
   turbulence 0/16
   scale <1/16,1/16,1/16>
   rotate <20,0,0>
   translate <0,0,0>
  }
#end

#macro tremie()
 union
 { union  // mur et paliers
  { box { <-1.20,-1.20*8/7, 0.10>, < 3.60, 1.20*13/7, 0.30> } // mur
   box { < 2.40,-1.20*8/7, 0.10>, < 3.60,-1.20*7/7, -0.95> } // palier -120
   box { <-1.20, 1.20*0/7, 0.10>, < 0.00,-1.20*1/7, -0.95> } // palier  000
   box { < 2.40, 1.20*6/7, 0.10>, < 3.60, 1.20*7/7, -0.95> } // palier +120
   texture
   { // pigment { color rgbt <1,0.9,0.8,0> }
    pigment_marbre()
    mon_finish(0.3) //0.3
   }
  }
  union  // glissieres
  { #local i=0; #while (i<8 )
    box { < 0.05 + 0.30*i, -1.20*8/7, 0.05>, < 0.25 + 0.30*i, 1.20*13/7, 0.10> }
   #local i=i+1; #end
   texture
   { pigment { color rgb <0,0,0> }
    mon_finish(0.8) // 0.8
   }
  }
```

```
 }
#end

#macro porte( ttt )
 object
 { polygon { 5, <0.00,0.00>, <0.90,0.00>, <0.90,0.90>, <0.00,0.90>, <0.00,0.00> }
  texture
  { pigment { color rgbt <1,0.9,0.8,0.8> } // clair bronze transparent
   mon_finish(0.3)      // reflechissant 30%
  }
  rotate <0,90,0>       //
  translate ttt       // position/etat de la porte
 }
#end

#macro marche( i, dh )
 object
 { union
  { box { <0,0,0>, <0.3,1.20/7,-0.90> }  // marche
   box { <0,0,0>, <0.3,1.20,0.10> }  // garde-corps sur glissiere
   scale <0.95,0.95,1>
   translate <0,-1.20/7,0>    // aligne le haut
  }
  texture
  { pigment { color rgb <0,0,0> }   // noir chrome
   mon_finish(0.8) // 0.8
  }
  translate <0.30*i, dh, 0>     // marches de 0 a 7
 }
#end

#macro volee( h0, h1 )  // les huit marches reliant les hauteurs h0 et h1
 #local i=0; #while (i<8)
  #local uu = i/7;
  marche( i, (1-uu)*h0 + uu*h1 )
 #local i=i+1; #end
#end

#macro roue( pos )
 object   // cylindre diametre 0.90, longueur 70cm, epaisseur 5cm
 { difference
  { cylinder { <0,0.45,-0.10> <0,0.45,-0.80> 0.45 }
   cylinder { <0,0.45,-0.05> <0,0.45,-0.85> 0.40 }
   translate pos      // position
  }
  texture
  { pigment { color rgb <0,1,0> }   // vert
   mon_finish(0)      // non reflechissant
  }
 }
#end

#macro personnage( pos, sens )
 object   // volume de 30/60/180
 { difference
  {
   box { < 0.00,0.00,-0.30> <0.30,1.80,0.30> }
   box { <-0.05,0.05,-0.35> <0.15,0.45,0.35> }
   box { <-0.05,1.50,-0.35> <0.35,1.85,-0.15> }
   box { <-0.05,1.50, 0.15> <0.35,1.85, 0.35> }
   scale <sens,1,1>
   translate <0,0,-0.45>
   translate pos      // position
  }
  texture
  { pigment { color rgb <0,0,1> }   // bleu
   mon_finish(0)      // non reflechissant
  }
```

```
 }
#end

#macro etat_volee( hA0, hA1, hB0, hB1, i, N ) // un etat de la volee pour i dans [0,1]
 #local uu = i/N;
 #local hA = (1-uu)*hA0 + uu*hA1;
 #local hB = (1-uu)*hB0 + uu*hB1;
 volee( hA, hB )
#end

#macro etat_roue( pos1, pos2, i, N )  // un etat de la roue pour i dans [0,1]
 #local uu = i/N;
 #local pos = (1-uu)*pos1 + uu*pos2;
 roue( pos )
#end

#macro etat_pers( pos1, pos2, sens, i, N ) // un etat du personnage pour i dans [0,1]
 #local uu = i/N;
 #local pos = (1-uu)*pos1 + uu*pos2;
 personnage( pos, sens )
#end

#macro animation( ii ) // ii dans [0,1]
 #local N = 8;  // il y a N etapes successives
 #local dN = 1/N;
 // [0,1]
 #if ( ii>= 0*dN & ii< 1*dN ) // la volée se déforme pour relier les niveaux -120 et 000
  #local i=0;
  etat_volee(-1.20, 0.00,-1.20,-1.20,      ii-i*dN, dN ) // variable
  etat_roue( <3.30, 1.20, 0.00>, <3.30, 1.20, 0.00>,   ii-i*dN, dN ) // attente
  etat_pers( <2.40,-1.20, 0.00>, <2.40,-1.20, 0.00>, 1, ii-i*dN, dN ) // fixe
  porte( <2.40,-1.20, 0.90> )     // porte -120 ouverte
  porte( <0.00, 0.00, 0.00> )     // porte  000 fermee
  porte( <2.30, 1.20, 0.00> )     // porte +120 fermee
 #end
 // [1,2]
 #if ( ii>= 1*dN & ii< 2*dN ) // le personnage se déplace de <2.40,-1.20,-0.45> à <0.00, 0.00,-0.45>
  #local i=1;
  etat_volee( 0.00, 0.00,-1.20,-1.20,      ii-i*dN, dN ) // fixe
  etat_roue( <3.30, 1.20, 0.00>, <3.30, 1.20, 0.00>,   ii-i*dN, dN ) // attente
  etat_pers( <2.40,-1.20, 0.00>, <0.00, 0.00, 0.00>, 1, ii-i*dN, dN ) // variable
  porte( <2.40,-1.20, 0.90> )     // porte -120 ouverte
  porte( <0.00, 0.00, 0.90> )     // porte  000 ouverte
  porte( <2.30, 1.20, 0.00> )     // porte +120 fermee
 #end
 // [2,3]
 #if ( ii>= 2*dN & ii< 3*dN ) // la volée se déforme pour relier les niveaux 000 et +120
  #local i=2;
  etat_volee( 0.00, 0.00,-1.20, 1.20,      ii-i*dN, dN ) // variable
  etat_roue( <3.30, 1.20, 0.00>, <3.30, 1.20, 0.00>,   ii-i*dN, dN ) // attente
  etat_pers( <0.00, 0.00, 0.00>, <0.00, 0.00, 0.00>,-1, ii-i*dN, dN ) // se retourne
  porte( <2.40,-1.20, 0.00> )     // porte -120 fermee
  porte( <0.00, 0.00, 0.00> )     // porte  000 fermee
  porte( <2.30, 1.20, 0.00> )     // porte +120 fermee
 #end
 // [3,4]
 #if ( ii>= 3*dN & ii< 4*dN ) // le personnage se déplace de <0.00, 0.00,-0.45> à <2.40, 1.20,-0.45>
  #local i=3;
  etat_volee( 0.00, 0.00, 1.20, 1.20,      ii-i*dN, dN ) // fixe
  etat_roue( <3.30, 1.20, 0.00>, <3.30, 1.20, 0.00>,   ii-i*dN, dN ) // attente
  etat_pers( <0.00, 0.00, 0.00>, <2.40, 1.20, 0.00>,-1, ii-i*dN, dN ) // variable
  porte( <2.40,-1.20, 0.00> )     // porte -120 fermee
  porte( <0.00, 0.00, 0.90> )     // porte  000 ouverte
  porte( <2.30, 1.20, 0.90> )     // porte +120 ouverte
 #end
 // [4,5]
 #if ( ii>= 4*dN & ii< 5*dN ) // la volée se déforme pour être horizontale à +120
  #local i=4;
```

```
  etat_volee( 0.00, 1.20, 1.20, 1.20,      ii-i*dN, dN ) // variable
  etat_roue( <3.30, 1.20, 0.00>, <3.30, 1.20, 0.00>,   ii-i*dN, dN ) // attente
  etat_pers( <2.40, 1.20, 0.00>, <2.85, 1.20, 0.00>,-1, ii-i*dN, dN ) // variable
  porte( <2.40,-1.20, 0.00> )     // porte -120 fermee
  porte( <0.00, 0.00, 0.00> )     // porte  000 fermee
  porte( <2.30, 1.20, 0.90> )     // porte +120 ouverte
 #end
 // [5,6]
 #if ( ii>= 5*dN & ii< 6*dN ) // la roue et le personnage passent sur la volée
  #local i=5;
  etat_volee( 1.20, 1.20, 1.20, 1.20,      ii-i*dN, dN ) // fixe
  etat_roue( <3.30, 1.20, 0.00>, <1.20, 1.20, 0.00>,   ii-i*dN, dN ) // variable
  etat_pers( <2.85, 1.20, 0.00>, <0.75, 1.20, 0.00>,-1, ii-i*dN, dN ) // variable
  porte( <2.40,-1.20, 0.00> )     // porte -120 fermee
  porte( <0.00, 0.00, 0.00> )     // porte  000 fermee
  porte( <2.30, 1.20, 0.90> )     // porte +120 ouverte
 #end
 // [6,7]
 #if ( ii>= 6*dN & ii< 7*dN ) // tout le monde descend de +120 à -120
  #local i=6;
  etat_volee( 1.20,-1.20, 1.20,-1.20,      ii-i*dN, dN ) // variable
  etat_roue( <1.20, 1.20, 0.00>, <1.20,-1.20, 0.00>,   ii-i*dN, dN ) // variable
  etat_pers( <0.75, 1.20, 0.00>, <0.75,-1.20, 0.00>,-1, ii-i*dN, dN ) // fixe
  porte( <2.40,-1.20, 0.00> )     // porte -120 fermee
  porte( <0.00, 0.00, 0.00> )     // porte  000 fermee
  porte( <2.30, 1.20, 0.00> )     // porte +120 fermee
 #end
 // [7,8]
 #if ( ii>= 7*dN & ii<= 8*dN ) // la roue et le personnage quittent la volée
  #local i=7;
  etat_volee(-1.20,-1.20,-1.20,-1.20,      ii-i*dN, dN ) // fixe
  etat_roue( <1.20,-1.20, 0.00>, <3.30,-1.20, 0.00>,   ii-i*dN, dN ) // passe de 120 a 330
  etat_pers( <0.75,-1.20, 0.00>, <2.85,-1.20, 0.00>,-1, ii-i*dN, dN ) // fixe
  porte( <2.40,-1.20, 0.90> )     // porte -120 ouverte
  porte( <0.00, 0.00, 0.00> )     // porte  000 fermee
  porte( <2.30, 1.20, 0.00> )     // porte +120 fermee
 #end
#end

#macro interpolation_lineaire( p, tt )
 (1-tt)*p[0] + tt*p[1]
#end

#macro interpolation_quadrique( p, tt )
 (1-tt)*(1-tt)*p[0] + 2*(1-tt)*tt*p[1] + tt*tt*p[2]
#end

#macro interpolation_cubique( p, tt )
 (1-tt)*(1-tt)*(1-tt)*p[0] + 3*(1-tt)*(1-tt)*tt*p[1] + 3*(1-tt)*tt*tt*p[2] + tt*tt*tt*p[3]
#end

#local cam = array[3] { <-1,0,-3>, <-1, 4,-2>, <-1,0,-2> }

// 2) SCENE :

light_source { <-3,2,-1> color rgb <1,1,1> }
light_source { < 1,2,-1> color rgb <1,1,1> }
background { color rgb <1/2,1/2,1/2> }
axes()
tremie()

//camera { location <-1/2,1,-1>*3 look_at <0.9,0.45,0> }
//camera { orthographic location <0,2,0> look_at <0,0,0> }


#local tt = 4/8; // clock; // i/N // choisir un multiple du nombre d'etapes
camera { location interpolation_quadrique( cam, tt )*2 look_at <0.9,0.45,0> }
animation( tt )
```