

## Code Processing vidéo de synthèse de Coding a moon

```
int randomColor=0,r=0,v=0,b=0,alpha=255;
float noiseCoef=random(0,1),valPulse=0;
float ALx=400, ALy=50, ALz=0; //ambient light coordinates
float angularSpeed=180; //angular speed for light path
float AML1x=0,AML1y=0,AML1z=0; //miniLune1 coordinates
float angularSpeedML1=0;
boolean j=true,k=true;
PImage luneTexture,miniLune1Texture,miniLune2Texture;
PShape lune;
PShape miniLune1;

void setup ()
{
  background (0);
  //fullScreen(P3D); // code pour être en plein écran, supprimer la ligne en dessous quand actif
  size (800,600,P3D);
  lights();
  noStroke();
  frameRate=30;
  //shapeMode(CENTER);
  luneTexture = loadImage("textureMetal.jpg");
  lune = createShape(SPHERE,100);
  lune.setTexture(luneTexture);

  miniLune1Texture = loadImage("textureBulles.jpg");
  miniLune1 = createShape(SPHERE,20);
  miniLune1.setTexture(miniLune1Texture);
  miniLune1Texture = loadImage("textureMetal.jpg");
}

void draw()
{
  background(0);
  if (frameCount%2==0)
  {
    if(j==true){noiseCoef=noiseCoef*1.05;}
    if(j==false){noiseCoef=noiseCoef*0.95;}
    if(noiseCoef>=0.9){j=false;}
    if(noiseCoef<=0.1){j=true;}
    randomColor = int(255*(noise(noiseCoef)));
  }

  ambientLight(0,0,40,200,200,0);

  angularSpeed=angularSpeed+0.2; // ici on change la vitesse de la rotation de la source de lumière autour du
  centre de la scène

  if(angularSpeed>360){angularSpeed=1;}

  // détermination des coordonnées de la source de lumière
  ALx=cos(radians(angularSpeed));
  ALy=sin(radians(angularSpeed));
  ALz=cos(radians(angularSpeed));
  directionalLight(255,255,randomColor,ALx,ALy,ALz);

  translate(width/2,height/2,20);
  // on les formes au centre de la scène, avec un décalage en Z de 20
```

```

// on dessine la planete centrale (appelée lune ici)
beginShape();
lune.rotateY(PI/1200); // on la fait tourner sur elle-même
shape(lune);
endShape();

// on dessine la mini-lune en rotation autour de la planete centrale
beginShape();
AML1x=3*cos(radians(angularSpeedML1)); // on fait varier les coordonnées X de la mini-lune
AML1z=3*sin(radians(angularSpeedML1)); // on fait varier les coordonnées Z de la mini-lune. Le *3 permet
d'élargir le cercle de la trajectoire
miniLune1.translate(AML1x,0,AML1z); // on applique la transformation des coordonnées x,y,z

translate(0,0,-50); // on place la lune au centre de la scène en tenant compte du diamètre de la planète centrale
(le -50)
// attention les fonctions translate s'accroissent (utiliser popMatrix pour éviter ça si besoin)
shape(miniLune1);
angularSpeedML1++;

if(angularSpeedML1==360)
{
angularSpeedML1=0;
miniLune1.resetMatrix();
}
endShape();
}

```