VII – Les objets 3D

101

VII.1 – Les objets « PShape » de Processing

- Les objets dans Processing : Forme (« shape »)
  - constituée de plusieurs sommets,
  - 2D ou 3D,
  - éventuellement texturée...
- Description d'un objet, de deux manières différentes :
  - Description par programme, avec des fonctions élémentaires :
    - `createShape(forme)` : formes connues (`ELLIPSE`, `RECT`, `ARC`, `TRIANGLE`, `SPHERE`, `BOX`, `QUAD`, `LINE`)
    - `beginShape()` : début de la description de l'objet (forme constituée de plusieurs sommets)
      `vertex(x,y,z)` : déclaration d'un sommet (à répéter pour chaque sommet de la forme)
      `endShape(CLOSE)` : fin de la forme avec tracé du dernier segment pour clore la forme
  - Chargement à partir d'un fichier (exemple avec un fichier de format SVG) :

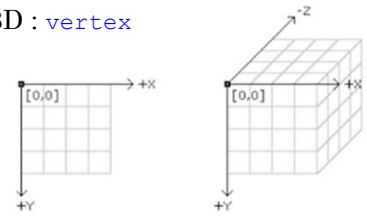
```
PShape s;
void setup() {
 s = loadShape("France.svg");
 smooth(); noLoop();
}
void draw() {
 shape(s,0,0,100,100);
}
```

102

VII.1 – La représentation géométrique des objets

- Coordonnées des sommets, en 2D ou 3D : `vertex`
  - Liste de (x,y), ou liste de (x,y,z)
  - Exemple pour un cube

```
// Face Avant du cube
monCube.vertex(-10,-10, 10);
monCube.vertex( 10,-10, 10);
monCube.vertex( 10, 10, 10);
...
```



- Coordonnées normalisées
  - normalisées : toutes dans l'intervalle [0,1], ou encore [-1,1]
  - puis mise à l'échelle par `scale` :

```
scale(s)
scale(x, y)
scale(x, y, z)
```

- Couleur de remplissage et de tracé :
  - `fill`, `noFill`, `setFill` et `stroke`, `noStroke`, `setStroke`

```
monCube.setFill(color(0, 255, 0)); // En Vert
monCube.setStroke(false);          // Sans trace du pourtour
```

103

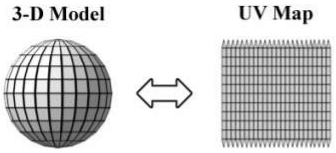
103

VII.1 – Principe de « mapping » des textures

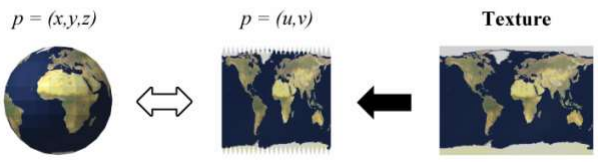
- Principe d'installation d'une texture sur un objet :
  - Texture = Image, au format standard

```
Texture_Cube = loadImage("ISIMA.jpg");    monCube.texture(Texture_Cube);
```

  - « UV map » :
    - Décrire la manière dont une texture 2D est appliquée à une surface 3D
    - UV : coordonnées horizontales et verticales



https://en.wikipedia.org/wiki/UV_mapping



- Choix de coordonnées normalisées ou non : `textureMode`
  - IMAGE ou NORMAL : coordonnées image, ou normalisées [0,1]

104

104

VII.2 – Description d'objets par programme (Résumé)

- Déclaration de l'objet PShape :

```
PShape monCubeV1;   // Un cube tout simple, de forme connue
PShape monCubeV2;   // Une autre forme, mais de type cube...
```

- Initialisation de l'objet : `createShape`

```
monCubeV1 = createShape(BOX,200); // C'est tout, car forme connue !
monCubeV2 = createShape(); // Forme quelconque, il faut preciser
                           // tous les points "sommets"
```

- Description de la géométrie de l'objet : `beginShape`

```
monCubeV2.beginShape(QUADS); // Quadrilateres entre les sommets
```

- Description (éventuelle) de la texture de l'objet : `texture`

```
monCubeV2.texture(maTexture); // maTexture est une image deja chargee
```

- Description effective de l'objet : `vertex`
  - Liste ici de tous les sommets de l'objet, avec éventuellement, la position UV de texture

```
monCubeV2.vertex(x, y, z, u, v);  // (x,y,z) = coordonnees du point 3D
                                  // (u,v) = coordonnees de la texture
```

  - Attention, à la fin, il faut clôturer la liste : `monCubeV2.endShape()`

105

105

VII.3 – Exemple 13 : Tracé de cubes, avec et sans texture (1/3)

```
...

// Declaration des variables globales
PShape Cube_Simple;  // Cube (shape) sans texture
PShape Cube_ISIMA;   // Le meme, avec une texture
PImage Texture_Cube; // Texture associee : une image JPG

// Fonction d'initialisation de l'application - executee une seule fois
void setup() {
  // Initialisation des parametres graphiques utilises
  size(640, 360, P3D); // Ouverture en mode 3D
  surface.setTitle("Exemple 8 - Shape : Cubes avec et sans texture - E. Mesnard / ISIMA");

  // Creation d'un cube tout simple
  fill(#71CE6C); stroke(#94DEA9); // des couleur unies, plutot vert pale
  Cube_Simple = createShape(BOX,width/3.5); // et de taille proportionnelle a la largeur

  // Creation d'un cube avec une texture
  Texture_Cube = loadImage("ISIMA.jpg"); // Chargement de la texture
  textureMode(NORMAL); // Les coordonnees de la texture sont normalisees de 0 a 1 en U et en V
  Cube_ISIMA = CreerCubeTexture(Texture_Cube,width/3.5);
}
```

106

106

VII.3 – Exemple 13 : Tracé de cubes, avec et sans texture (2/3)

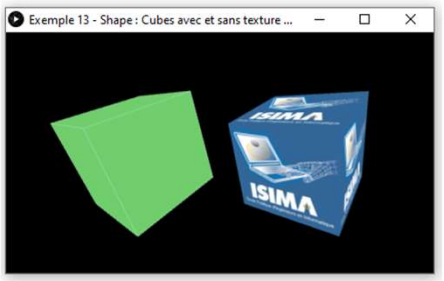
```
// Fonction de re-tracage de la fenetre - executee en boucle
void draw() {
  // Effacer la fenetre avant de dessiner
  background(0);

  // Positionnement des cubes et tracages...
  translate(width/4.0, height/2.0, -100);
  shape(Cube_Simple);
  translate(width/2.0, 0, 0); // Attention : les deplacements sont cumulatifs
  shape(Cube_ISIMA); // Trace effectif de ce cube
}

// Fonction de creation d'un cube, avec une taille et une texture
PShape CreerCubeTexture(PImage tex, float taille) {
PShape formeCube;

  formeCube = createShape();
  formeCube.beginShape(QUADS);
  formeCube.noFill();     // Parametres graphiques
  formeCube.noStroke();
  formeCube.texture(tex); // A definir des le depart
```

107

107

VII.3 – Exemple 13 : Tracé de cubes, avec et sans texture (3/3)

```
  // Declaration des listes de points pour les faces
  // vertex(x, y, z, u, v) : (x,y,z) = coordonnees du point
  //                         (u,v) = coordonnees de la texture

  // Face +Z = Avant
  formeCube.vertex(-1,-1, 1, 0, 0);
  formeCube.vertex( 1,-1, 1, 1, 0);
  formeCube.vertex( 1, 1, 1, 1, 1);
  formeCube.vertex(-1, 1, 1, 0, 1);

  // Face -Z = Arriere
  formeCube.vertex( 1,-1,-1, 0, 0);
...
  formeCube.vertex(-1, 1,-1, 0, 1);

  formeCube.endShape();

  // Mise a l'echelle
  formeCube.scale(taille/2.0); // taille proportionnelle

  // Orientation initiale possible
  // formeCube.rotateX(PI/6);
  // formeCube.rotateY(PI/4);
  return formeCube;
}
```

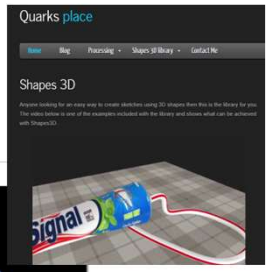


108

108

VII.4 – Extension avec bibliothèques de gestion d'objets 3D

- Bibliothèque 3D : « Shapes 3D for Processing » de Peter Lager (« Quark »)
  - Directement depuis Processing, ou bien, sur son site : http://www.lagers.org.uk
    http://www.lagers.org.uk/s3d4p/libguides/lg01-introduction.html



- Exemple 13bis du cube texturé, en quelques lignes...

```
import shapes3d.*; // Bibliotheque trace 3D
Box Cube_ISIMA;    // Variable shapes3D : un cube !
PImage Ma_Texture; // Texture associee

void setup() {
  size(400,400,P3D);
  // Creation du CUBE 3D en objet SHAPES 3D
  Ma_Texture = loadImage("ISIMA.jpg");
  Cube_ISIMA = new Box(120,120,120);
  Cube_ISIMA.texture(Ma_Texture);
  Cube_ISIMA.drawMode(Shape3D.TEXTURE);
}

void draw() {
  background(0);
  Cube_ISIMA.moveTo(width/2, height/2, 65);
  Cube_ISIMA.rotateBy(0.05,0.02,0); // on le fait tourner un peu...
  Cube_ISIMA.draw(getGraphics());   // dessin sur la fenêtre principale
}
```

109

109

VII.5 – Description d'objets par fichier

- Principe :
  - Dessiner les objets texturés complexes dans des logiciels dédiés...
  - ... puis importer les objets dans Processing !
- Outil logiciel (gratuit) le plus complet : Blender
  - Liens :
    https://www.blender.org/download/
    https://openclassrooms.com/courses/debutez-dans-la-3d-avec-blender

110

110

VII.5 – Description d'objets 3D Wavefront par fichier OBJ

- Objets Wavefront :
  - Liens : https://fr.wikipedia.org/wiki/Objet_3D_(format_de_fichier)
    https://en.wikipedia.org/wiki/Wavefront_.obj_file
  - Objets 3D, statiques (maillages statiques : « static meshes »)
  - Décrits par 1, 2 ou 3 fichiers :
    - OBJ (formes de l'objet), MTL (« material », couleur) et JPG (texture, appelée dans le mtl)

```
#  Fichier OBJET d'un dragon
mtllib ./texture.mtl
g
v -2.809412 -1.633775 2.582798
v -2.708658 -1.155276 2.354352
v -2.966226 -1.194807 2.692873
...
g body
usemtl lcdgreen
s 1
f 307 306 310 311
f 305 291 308 309
f 298 295 306 307
...
```

```
newmtl white
Ka 0.4000 0.4000 0.4000
Kd 1.0000 1.0000 1.0000
Ks 0.3000 0.3000 0.3000
illum 2
Ns 60.0000

newmtl red
Ka 0.4449 0.0000 0.0000
...
newmtl my_texture_from_jpg
illum 4
Kd 1.00 1.00 1.00
Ka 1.00 1.00 1.00
Tf 1.00 1.00 1.00
map_Kd texture5.jpg
```

  - Beaucoup d'exemples en téléchargement libre (site `free3D.com`, ...)

111

111

VII.5 – Description d'objets par fichier (Résumé)

- Déclaration de l'objet Shape :

```
PShape monObjet;   // Un objet, decrit dans un fichier OBJ
```

- Initialisation et chargement de l'objet : `loadShape`

```
monObjet = loadShape("fichier_objet.obj"); // Chargement tout simple...
```

- Récupération (éventuelle) de la géométrie de l'objet : `get...`

```
tailleX = monObjet.getWidth();  // Dimension 3D de mon objet
tailleY = monObjet.getHeight(); // telle que decrite dans
tailleZ = monObjet.getDepth();  // le fichier OBJ
```

- Mise (éventuelle) à l'échelle souhaitée : `scale`

```
monObjet.scale(facteur_agrandissement); // Idem sur les 3 directions
```

- Affichage effectif de l'objet : `shape`

```
lights();          // Allumage de la lumiere afin d'avoir
shape(monObjet);   // des calculs d'ombre projetee sur mon objet 3D
```

112

112

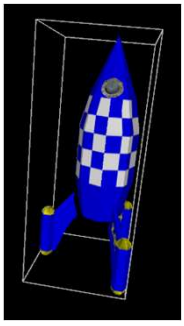
VII.6 – Exemple 14 : Import d'un objet 3D OBJ (1/2)

```
/*****************************************************************************/
/*  MESNARD Emmanuel                                                 ISIMA   */
/*                                                                           */
/*          Exemple 14 : Dessin d'un objet OBJ texture par P3D               */
/*                                                                           */
/* Exemple_14_Dessin_OBJ.pde                              Processing 3.0     */
/*****************************************************************************/

// Declaration des variables globales
PShape objetOBJ;        // Objet a charger
PShape boundingBoxOBJ;  // Boite englobante
float dimOBJ_X,dimOBJ_Y,dimOBJ_Z; // Dimensions de l'objet

// Fonction d'initialisation de l'application - executee une seule fois
void setup() {
  size(640,480,P3D); // P3D obligatoire pour le rendu des OBJ
  surface.setTitle("Exemple 14 - Dessin d'objet OBJ - E. Mesnard / ISIMA");

  // Chargement de l'objet utilise
  objetOBJ = loadShape("Dog_Isima.obj"); // Objet avec texture

  // Recuperation des dimensions de l'objet
  dimOBJ_X = objetOBJ.getWidth();
  dimOBJ_Y = objetOBJ.getHeight();
  dimOBJ_Z = objetOBJ.getDepth();
```

113

VII.6 – Exemple 14 : Import d'un objet 3D OBJ (2/2)

```
  // Calcul du facteur d'echelle pour affichage complet a l'ecran
  facteurEchelle = min(width/(2.0*dimOBJ_X),height/(2.0*dimOBJ_Y),width/(2.0*dimOBJ_Z));
  objetOBJ.scale(facteurEchelle);
  // Mise a jour des dimensions de l'objet
  dimOBJ_X *= facteurEchelle; dimOBJ_Y *= facteurEchelle; dimOBJ_Z *= facteurEchelle;
  // Attention : il faut souvent appliquer une translation pour recentrer l'objet...
  // Par exemple, si l'origine de l'objet est en bas a gauche au fond, il faut appliquer :
  objetOBJ.translate(dimOBJ_X/2, -dimOBJ_Y/2, -dimOBJ_Z/2);

  // Creation d'une boite englobante
  noFill(); stroke(200); // Pourtour gris clair, sans remplissage
  boundingBoxOBJ = createShape(BOX,dimOBJ_X,dimOBJ_Y,dimOBJ_Z);

  // Trace de cette boite par defaut
  traceBB_OBJ = true;
} // fin de setup

// Fonction de re-tracage de la fenetre - executee en boucle
void draw() {
  background(0); // Effacer la fenetre avant de dessiner
  translate(width/2, height/2, 0);  // Positionnement au centre de la fenetre
  lights();       // Affichage des objets... avec un peu de lumiere !
  shape(objetOBJ);
  if (traceBB_OBJ) shape(boundingBoxOBJ);
}
```



114