STRAINCHAMPS Clothilde
CORNEVIN Camille

Rapport Synthèse d'image

Le Lapin

I- Conception

1- Commencement

Avant de commencer la réalisation de notre lapin, nous avons cherché de l'inspiration avec des lapins de dessins animés trouvés sur internet. Nous en avons trouvé un qui nous plaisait mais nous avons finalement décidé de ne pas nous inspirer d'un modèle pour être plus libres dans notre réalisation.

2- La tête du lapin

Nous avons commencé par réaliser la tête du lapin. Pour cela nous avons fait une sphère légèrement étirée pour que la tête ne soit pas tout à fait une boule.

```
glScalef(1, 0.90, 1); //sphère étirée en hauteur
glutSolidSphere(4,50, 50); //sphère de rayon 4
```

La tête

Pour les joues nous avons pensé à les faire ressortir en utilisant deux autres sphères 2.6 fois plus petites que la tête.

```
glPushMatrix();
glTranslatef(-2,-1.5,2); //translation pour placer la sphère en bas à gauche de la tête
glutSolidSphere(1.5,50, 50); //sphère de rayon 1.5
glPopMatrix();

glPushMatrix();
glTranslatef(2,-1.5,2); //translation pour placer la sphère en bas à gauche de la tête
glutSolidSphere(1.5,50, 50);
glPopMatrix();
```

Les joues

Nous avons réalisé un nez le plus ressemblant possible au museau presque triangulaire d'un lapin. Pour cela nous lui avons fait une sphère étirée en ovale accolée à une autre sphère juste en dessous.

```
glPushMatrix();
glTranslatef(0,-0.5,4); //avancement de 4 pour le faire ressortir
glScalef(1.5, 0.75, 0.75); //aplatissement pour une forme ovale en longueur
glutSolidSphere(1,50, 50); //sphère de rayon 1
glPopMatrix();

glPushMatrix();
glTranslatef(0,-0.75,4);//sphère légèrement plus basse
glutSolidSphere(1,50, 50);
glPopMatrix();
```

Le nez

Les dents du lapin sont deux cubes allongés et aplatis légèrement écartés l'un de l'autre pour un effet plus réaliste.

```
glPushMatrix();
glTranslatef(-0.2,-2,3.5); //on place la dent juste en dessous du nez, légèrement décalée à gauche
glScalef(0.5, 1, 0.5); //redimensionnement du cube pour avoir une belle forme allongée
glutSolidCube(0.75); // cube de 0.75
glPopMatrix();

glPushMatrix();
glTranslatef(0.2,-2,3.5); //on place la dent juste en dessous du nez, légèrement décalée à droite
glScalef(0.5, 1, 0.5);
glutSolidCube(0.75);
glPopMatrix();
```

Les dents

Les yeux sont, comme les joues, deux sphères mais qui ressortent légèrement moins cette fois. Par la suite nous avons intégrées deux plus petites sphères à l'intérieur des précédentes pour représenter les pupilles.

```
glPushMatrix();
glTranslatef(-1.5,1.5,2.5);//œil décalé en haut à gauche et avancé pour ressortir
glutSolidSphere(1,50, 50);  //sphère de rayon 1
glPopMatrix();

glPushMatrix();
glTranslatef(1.5,1.5,2.5); //œil décalé en haut à droite et avancé pour ressortir
glutSolidSphere(1,50, 50); //sphère de rayon 1
glPopMatrix();


//Pupilles

glPushMatrix();
glTranslatef(-1.6,1.6,3.2); // à l'intérieur de l'œil gauche et avancé pour ressortir légèrement
glutSolidSphere(0.5,50, 50); //sphère de rayon 0.5
glPopMatrix();

glPushMatrix();
glTranslatef(1.6,1.6,3.2); // à l'intérieur de l'œil droit et avancé pour ressortir légèrement
glutSolidSphere(0.5,50, 50); //sphère de rayon 0.5
glPopMatrix();
```

Les yeux

Pour les oreilles de notre lapin, nous avions tout d'abord essayé de les réaliser avec deux cônes mais le rendu n'était pas du tout esthétique et encore moins réaliste. Nous avons donc opté pour des sphères que nous avons ovalisées.

```
//Oreille droite

glPushMatrix();
glTranslatef(1.5,3.5,0);  //on place l'oreille à droite
glRotatef(10, 0, 0, -1);//orientation de l'oreille
glRotatef(10, 0, 1, 0); //orientation de l'oreille
glScalef(0.25, 1.5, 0.1);  //aplatissement et allongement
glutSolidSphere(4, 50, 50);  //sphère de rayon 4
glPopMatrix();

//Oreille gauche

glPushMatrix();
glTranslatef(-1.5,3.5,0); //on place l'oreille à droite
glRotatef(30, 0, 0, 1); //orientation de l'oreille
glRotatef(30, 0, -1, 0); //orientation de l'oreille
glScalef(0.25, 1.5, 0.1); //aplatissement et allongement
glutSolidSphere(4, 50, 50); //sphère de rayon 4
glPopMatrix();
```

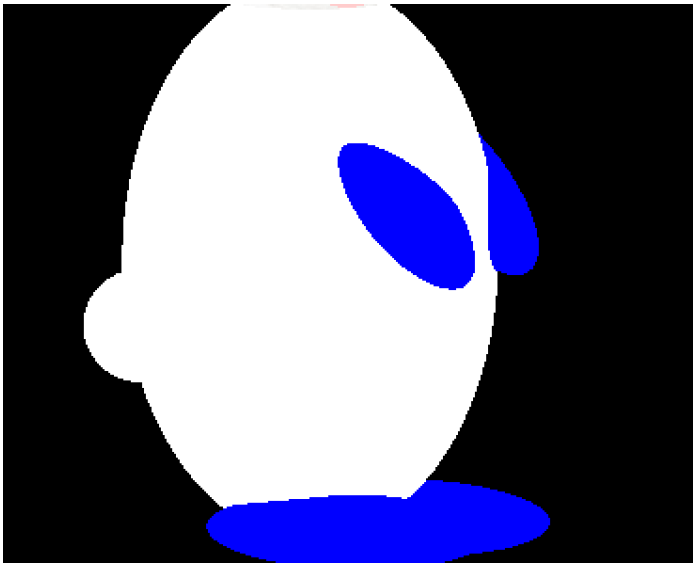
Les oreilles

Les moustaches sont également des cubes allongés et aplatis.

```
//Moustaches droite


glPushMatrix();
glTranslatef(1.20,-1.4,4.5);// on la place à droite du nez
glScalef(2, 0.05, 0.05);  //on allonge et aplati
glutSolidCube(1.25);  //cube de 1.25
glPopMatrix();

glPushMatrix();
glTranslatef(1.15,-1.55,4.5);
glRotatef(10, 0, 0, -1);  //on pivote de 10°
glScalef(2, 0.05, 0.05);
glutSolidCube(1.25);
glPopMatrix();

glPushMatrix();
glTranslatef(1.15,-1.25,4.5);
glRotatef(10, 0, 0, 1);
glScalef(2, 0.05, 0.05);
glutSolidCube(1.25);
glPopMatrix();
```

La moustache

3- Le corps du lapin



```
glTranslatef(0, -10, 0);//on la place en dessous de la tête
glScalef(1, 1.5, 1); //on l'allonge légèrement en hauteur
glutSolidSphere(5, 50, 50); //sphère de rayon 5
```

Le corps

Les pattes avant sont deux sphères ovalisées et inclinées pour ne ressortir qu'en parti du haut du corps.

```
glPushMatrix();
glTranslatef(-2,-8,3.5);
glRotatef(40, -1, 0, 0); //on incline
glScalef(1,1.5,0.5);
glutSolidSphere(2,50,50);
glPopMatrix();

glPushMatrix();
glTranslatef(2,-8,3.5);
glRotatef(40, -1, 0, 0);
glScalef(1,1.5,0.5);
glutSolidSphere(2,50,50);
glPopMatrix();
```

Les pattes avant

```
glPushMatrix();
glTranslatef(-2,-17,2); //on place les pattes en dessous du corps
glScalef(1,0.5,2);
glutSolidSphere(2,50,50);
glPopMatrix();

glPushMatrix();
glTranslatef(2,-17,2);
glScalef(1,0.5,2);
glutSolidSphere(2,50,50);
glPopMatrix();
```

Les pattes arrière

```
glTranslatef(0,-12,-5); //on la place à l'arrière du corps
glutSolidSphere(1.5,50,50); //sphère de rayon 1.5
```

La queue

4- La cape du lapin

```
glPushMatrix();
glTranslatef(1,-7.2,-6.2); //on place au milieu de la cape
glRotatef(40, 1, 0, 0); //on incline pour que ça soit sur la cape
glScalef(1, 4, 0.05); //on allonge pour faire la partie verticale
glutSolidCube(1); //cube de 1
glPopMatrix();

glPushMatrix();
glTranslatef(0,-8.5,-7.2);
glRotatef(40, 1, 0, 0);
glScalef(3, 1, 0.05); //on allonge pour faire la partie horizontale
glutSolidCube(1);
glPopMatrix();
```

L sur la cape

```
glPushMatrix();
glTranslatef(0,-4.2,0); //on place au niveau du cou
glScalef(1.8, 0.01, 1.8); //on aplati pour l'effet ficelle
glutSolidSphere(2,50,50); //sphère pour le tour de cou
glPopMatrix();

glPushMatrix();
glTranslatef(0,-7.5,-6); //on place dans le dos
glRotatef(40, 1, 0, 0); //on l'incline de 40°
glScalef(1.5, 3, 0.05); //on aplati
glutSolidCube(3); //cube de base de 3
glPopMatrix();

glPushMatrix();
glTranslatef(2.3,-7.5,-6);//partie de droite
glRotatef(40, 1, 0, 0);
glRotatef(45, 0, 0, 1); //on pivote pour faire un losange
glScalef(1, 1, 0.05);
glutSolidCube(6.3);
glPopMatrix();

glPushMatrix();
glTranslatef(-2.3,-7.5,-6); //partie de gauche
glRotatef(40, 1, 0, 0);
glRotatef(45, 0, 0, -1); //on pivote pour faire un losange
glScalef(1, 1, 0.05);
glutSolidCube(6.3);
glPopMatrix();

glPushMatrix();
glTranslatef(0,-9.2,-7.4); //partie basse
glRotatef(40, 1, 0, 0);
glScalef(3, 1, 0.05);
glutSolidCube(4.5);
glPopMatrix();
```

La cape

II- Mouvement

1- Mouvement continu

Les moustaches du lapin bougent en continu selon un angle de rotation par rapport à l'axe y. On les fait bouger en avant puis en arrière. Pour cela, on utilise la fonction *idle()* qui dans *glut* sert à effectuer des tâches de manière continue pendant l'exécution du programme.

```
void idle() {
    if (direction == 1) { //sens de rotation
        angleMoustache += 1.0;
        if (angleMoustache > 30.0)
            direction = -1;
    } else {
        angleMoustache -= 1.0;
        if (angleMoustache < -5.0)
            direction = 1;
    }
    glutPostRedisplay(); // demander un rafraîchissement de l'affichage
}
```

2- Mouvement contrôlé

On peut bouger les oreilles du lapin selon un angle de rotation par rapport à l'axe z. Les pointes des oreilles peuvent s'approcher en appuyant sur la touche « O » de notre clavier, et s'éloigner en appuyant sur la touche « o ». Pour cela, on rajoute un cas au *switch* de la fonction *clavier()* qui, quand on appuie sur une touche bien définie, réalise une action qui lui est associée.

```
case 'o' :
    if(angleOreille<=40) {
        angleOreille += 1;
        glutPostRedisplay();
    }
    break;
case 'O' :
    if(angleOreille>=0) {
        angleOreille -= 1;
        glutPostRedisplay();
    }
    break;
```

III- Texture

Pour notre projet, nous avons choisi deux textures. L'une sera un tissu en satin doré pour la cape et l'autre une fourrure pour le corps et la tête de notre lapin. Ces textures sont stockées dans un tableau de textures que l'on initialise au début de notre programme. Puis dans la fonction *main()* de notre programme, on paramètre nos textures. Voici un exemple pour la texture de la cape.

```
glGenTextures(2, texObject); // génère un nouvel objet de  texture

// Lier la texture 1 qui est le tissu pour la cape
glBindTexture(GL_TEXTURE_2D, texObject[0]);
// Paramètres de la texture 1
glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_MAG_FILTER,GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_MIN_FILTER,GL_NEAREST);
glTexImage2D(GL_TEXTURE_2D,0,GL_RGB,largimg1,hautimg1,0,
             GL_RGB,GL_UNSIGNED_BYTE,image1);
```

Paramétrage des textures

On fait de même pour la texture de la fourrure.

1- Texture aplatie

Nous avons appliqué une texture sur la cape, qui est un tissu en satin doré. La cape étant composée de cubes aplatit, il nous reste à appliquer sur les faces visibles notre texture. Voici un exemple pour l'un des cubes de la cape.

```
glEnable(GL_TEXTURE_2D); //on active la texture
glBindTexture(GL_TEXTURE_2D, texObject[0]); // on choisit notre
texture dans notre tableau de texture
glBegin(GL_POLYGON); //on commence l'appareillage d'un sommet de notre
image avec un sommet de notre cube
glTexCoord2f(0.0, 0.0); glVertex3f(-1.5, -1.5, -1.5);
glTexCoord2f(0.0, 1.0); glVertex3f(-1.5, 1.5, -1.5);
glTexCoord2f(1.0, 1.0); glVertex3f(1.5, 1.5, -1.5);
glTexCoord2f(1.0, 0.0); glVertex3f(1.5, -1.5, -1.5);
glEnd();
```

Texture sur les cubes de la cape

2- Texture enroulée

Nous avons enroulé une texture sur les parties du corps poilus du lapin, c'est-à-dire sur le corps, la tête, les oreilles et la queue. Pour enrouler notre texture sur une sphère, il nous a fallu construire cette sphère face par face avec des quadrilatères.

Quant à la texture, elle est enroulée autour de la sphère de manière linéaire, créant un motif continu sur la surface de la sphère en fonction des coordonnées de texture définies pour chaque quadrilatère.

Nous avons donc créé une fonction *sphereTexture().*

```
void sphereTexture(float r, int nbM, int nbP) {
    for (int i = 0; i < nbM; i++) {
        glBegin(GL_QUADS);
        for (int j = 0; j <= stacks; j++) {

      //on dessine tous les points d'un des quadrilatères
            float theta1 = (i * 2 * M_PI) / nbM;
            float phi1 = (j * M_PI) / nbP;
            float x1 = r * sin(phi1) * cos(theta1);
            float y1 = r * sin(phi1) * sin(theta1);
            float z1 = r * cos(phi1);

      ...

      //on applique la texture sur le quadrilatère fait

      //pour le point en haut à gauche (coin inférieur gauche de la texture)
glNormal3f(x1, y1, z1);
glTexCoord2f((float)i / nbM, (float)j / nbP);
glVertex3f(x1, y1, z1);

      ...
```

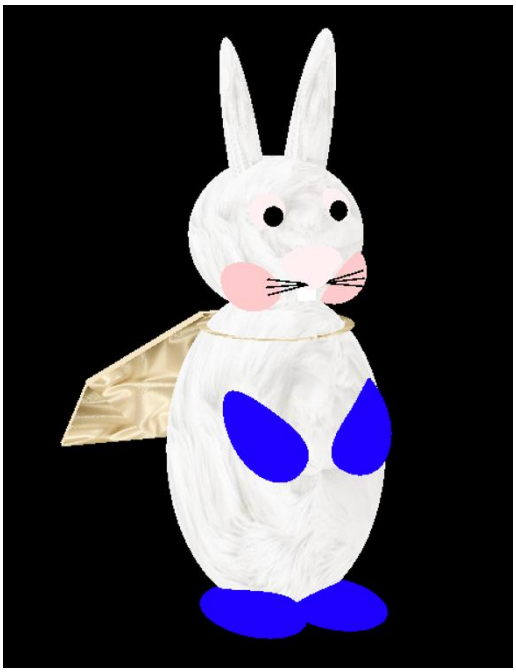
Fonction qui dessine une sphère et enroule la texture dessus

On peut utiliser notre fonction comme cela :

```
glPushMatrix();
glTranslatef(0, -10, 0);
glScalef(1, 1.5, 1);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, texObject[1]);
sphereTexture(5.0, 50, 50);
//glutSolidSphere(5,50, 50);
glDisable(GL_TEXTURE_2D);
glPopMatrix();
```

Pour le corps

Voici notre lapin complet avec les textures ajoutées :



IV- Camera

Pour ce projet une gestion de la caméra a aussi été implémentée. Dans le sujet il nous était demandé de tourner la caméra autour de notre lapin avec les flèches de notre clavier ce qui est normalement possible en appelant les objets **GLUT_KEY_LEFT, GLUT_KEY_RIGHT, GLUT_KEY_UP** et **GLUT_KEY_DOWN** mais sur certaines machines comme les Mac l'association n'est pas bonne. Par exemple **GLUT_KEY_LEFT** correspond à la touche « d » du clavier. Nous avons donc choisi de changer les touches.

Nous avons donc les touches suivantes pour modifier la vue :

- « d » la caméra tourne autour du lapin par la gauche
- « q » la caméra tourne autour du lapin par la droite
- « z » la caméra tourne autour du lapin par le bas
- « s » la caméra tourne autour du lapin par le haut

Pour pouvoir réaliser ces actions, il faut pouvoir modifier les caractéristiques de la fonction *gluLookAt()* au cours de l'exécution de notre programme. Nous le ferons par le biais de la fonction *clavier().*

Nous allons faire en sorte que les points de vue appartiennent au cercle de rayon 5 et de centre (0,0) en fonction de x et y pour tourner autour du lapin de droite à gauche et de gauche à droite. On fait de même pour de bas en haut et de haut en bas, mais en fonction de z et y.

```
// point de vue de l'utilisateur
gluLookAt(rayonCamera*cos(angleCamera)* cos(angleCamera2)*zoom,
rayonCamera*sin(angleCamera2)*zoom, rayonCamera*sin(angleCamera)*
cos(angleCamera2)*zoom, // Position de la caméra
      0.0, 0.0, 0.0, // Point de vue (regardé vers l'origine)
      0.0, 1.0, 0.0); // Vecteur d'orientation vers le haut (l'axe y)
```

Ensuite, pour le zoom, il suffit d'ajouter un facteur de zoom à notre formule. Cela est, en fait, le rayon du cercle d'observation. Il nous faut juste vérifier que, pour zoomer, le rayon doit être supérieur à 1. La touche « A » nous permet de zoomer et « a » de dézoomer.

V- Lumière

Pour la lumière, nous avons choisi deux lumières. L'une sera une lumière plutôt générale et l'autre sera comme un projecteur sur une scène. Pour cela, on définit nos lumières dans notre fonction *main()* comme ceci :

```
GLfloat mat_specular[] = { 0.3, 0.3, 0.3, 1.0 };
GLfloat mat_shininess[] = { 10.0 };
GLfloat light_position[] = { 0.0, 1.0, 0.0, 1.0 };
GLfloat light_ambient[] = {1.0f, 1.0f, 1.0f, 1.0f};
// Lumière ambiante
GLfloat light_diffuse[] = {1.0f, 1.0f, 1.0f, 1.0f};
// Lumière diffuse
GLfloat light_specular[] = {1.0f, 1.0f, 1.0f, 1.0f};
// Lumière spéculaire

glClearColor (0.0, 0.0, 0.0, 0.0);
glShadeModel (GL_SMOOTH);

glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
glLightfv(GL_LIGHT0, GL_POSITION, light_position);
glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular);
```

Première lumière (lumière générale)

```
glClearColor (0.0, 0.0, 0.0, 0.0);
glShadeModel (GL_SMOOTH);
glLightfv(GL_LIGHT1, GL_POSITION, light_position);
glLightf(GL_LIGHT1, GL_SPOT_CUTOFF, 45.0);
glLightf(GL_LIGHT1, GL_SPOT_EXPONENT, 2.0);
```

Deuxième lumière (projecteur)

La première lumière s'active en appuyant sur la touche « l » et se désactive en appuyant sur la touche « L ».

De même la deuxième lumière s'active en appuyant sur la touche « m » et se désactive en appuyant sur la touche « M ».

La deuxième lumière n'est pas vraiment fonctionnelle.

Voici notre lapin avec la première lumière