UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

INGENIERÍA EN COMPUTACIÓN

LABORATORIO DE COMPUTACIÓN GRÁFICA GRUPO 05

GRUPO DE TEORÍA:

SEMESTRE 2018-1

Proyecto final

Fecha de entrega límite: 30 de noviembre de 2017

Nº de Cuenta: 309143147

Email: woodynava93@gmail.com

Nombre completo:

Nava García Mario Guillermo

# Sección técnica:

Punto 1:

Se creó una primitiva para los 3 flippers y se instanció de las 3 maneras requeridas.

void Flippers() {

//Flippers

glPushMatrix();//Derecho

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, redDiffuseMaterial);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, whiteSpecularLight);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, blackAmbientLight);

glMaterialfv(GL\_FRONT, GL\_SHININESS, mShininess);

glTranslatef(3.0, -0.55, 12.95);

glRotatef(10, 1, 0, 0);

glRotatef(DerFlip, 0, 1, 0);

glScalef(1.0, 0.5, 3.0);

fig\_flippers.Flippers ( 0, flipper.GLindex);

glPopMatrix();

glPushMatrix();//Izquierdo

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, redDiffuseMaterial);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, whiteSpecularLight);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, blackAmbientLight);

glMaterialfv(GL\_FRONT, GL\_SHININESS, mShininess);

glTranslatef(-3.5, -0.55, 12.95);

glRotatef(10, 1, 0, 0);

glRotatef(IzqFlip, 0, 1, 0);

glScalef(1.0, 0.5, 3.0);

fig\_flippers.Flippers(0, 0);

glPopMatrix();

glPushMatrix();

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, redDiffuseMaterial);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, whiteSpecularLight);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, blackAmbientLight);

glMaterialfv(GL\_FRONT, GL\_SHININESS, mShininess);

glTranslatef(4.75, 0.15, 8.0);

glRotatef(10, 1, 0, 0);

glRotatef(CentFlip, 0, 1, 0);

glScalef(1.0, 0.5, 2.0);

fig\_flippers.Flippers(0, 0);

glPopMatrix();

}

Punto 2:

Palanca con resorte que se contrae y se activa con el mouse

void Resorte() {

//Resorte

glPushMatrix();

glTranslatef(7, -0.55, 12.25);

glRotatef(100, 1, 0, 0);

glRotatef(-90, 0, 1, 0);

glRotatef(180, 1, 0, 0);

glTranslatef(0.0, -2.0, 0);

fig\_resorte.cilindro(0.5, comp, 30, resorte.GLindex);

glPopMatrix();

//Palanca

glPushMatrix();

glTranslatef(7, -0.55, 12.25);

glRotatef(100, 1, 0, 0);

glRotatef(-90, 0, 1, 0);

glRotatef(180, 1, 0, 0);

glTranslatef(0.0, comp1, 0.0);

fig\_resorte.cilindro(0.5, 2.0, 30, azul.GLindex);

glPopMatrix();

glPushMatrix();

glTranslatef(7, -0.55, 12.25);

glRotatef(100, 1, 0, 0);

glRotatef(-90, 0, 1, 0);

glRotatef(180, 1, 0, 0);

glTranslatef(0.0, comp1, 0.0);

fig\_resorte.cilindro(1.0, 0.15, 30, azul.GLindex);

glPopMatrix();

}

void ControlRaton(int button, int state, int x, int y) {

if (button==GLUT\_LEFT\_BUTTON || state==GLUT\_DOWN) {

if (comp > 0.5) {

comp -= 1.5;

comp1 -=1.0;

}

zPos1 += 1.3;

}

if (button==GLUT\_LEFT\_BUTTON && state==GLUT\_UP) {

comp +=1.5;

comp1 +=1.0;

g\_fanimacion ^= true; //Activamos/desactivamos la animacÃon

}

if (button==GLUT\_RIGHT\_BUTTON && state==GLUT\_UP) {

if (IzqFlip < -60) {

IzqFlip +=60;

}

}

glutPostRedisplay();

}

Punto 3:

Se creó la función creadora del modelo jerárquico y se instanció 2 veces en los lugares deseados.

void jerarquico() {

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, CirculoDiffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, CirculoSpecular);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, CirculoAmbient);

glMaterialfv(GL\_FRONT, GL\_SHININESS, CirculoShininess);

glPushMatrix();//punta brazos

glTranslatef(0.0, 6.0, 0.0);

glTranslatef(0.0, 0, 0.0);

glRotatef(0, 1.0, 0.0, 0.0);

glRotatef(0, 0.0, 1.0, 0.0);

glRotatef(0, 0.0, 0.0, 1.0);

glPushMatrix();//Brazo 1

glTranslatef(0.35f, 0.2f, 0.0f);

glRotatef(-45, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix();//Brazo 1.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(-15, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix(); //Brazo 1.1.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(-30, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glScalef(2.5, 0.25, 0.25);

glColor3f(1.0, 0.2, 0.0);

prisma();

glPopMatrix(); // Sale del stack del Brazo 1.1.1

glScalef(2.5, .25, 0.25);

glColor3f(1.0, 1.0, 0.0);

prisma();

glPopMatrix();//Sale stack brazo 1.1

glScalef(2.5, 0.25, 0.25);

glRotatef(180, 1, 0, 0);

glColor3f(1.0, 0.0, 0.0);

prisma();

glPopMatrix();//Sale Brazo 1

glPushMatrix();//Brazo 2

glTranslatef(0.0f, 0.2f, -0.35f);

glRotatef(90, 0.0, 1.0, 0.0);

glRotatef(movBrazo1, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix();//Brazo 2.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo11, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix(); //Brazo 2.1.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo111, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glScalef(2.5, 0.25, 0.25);

glColor3f(1.0, 0.2, 0.0);

prisma();

glPopMatrix(); // Sale del stack del Brazo 2.1.1

glScalef(2.5, .25, 0.25);

glColor3f(1.0, 1.0, 0.0);

prisma();

glPopMatrix();//Sale stack brazo 2.1

glScalef(2.5, 0.25, 0.25);

glRotatef(180, 1, 0, 0);

glColor3f(1.0, 0.0, 0.0);

prisma();

glPopMatrix();//Sale Brazo 2

glPushMatrix();//Brazo 3

glTranslatef(-0.35f, 0.2f, 0.0f);

glRotatef(180, 0.0, 1.0, 0.0);

glRotatef(movBrazo1, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix();//Brazo 3.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo11, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix(); //Brazo 3.1.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo111, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glScalef(2.5, 0.25, 0.25);

glColor3f(1.0, 0.2, 0.0);

prisma();

glPopMatrix(); // Sale del stack del Brazo 3.1.1

glScalef(2.5, .25, 0.25);

glColor3f(1.0, 1.0, 0.0);

prisma();

glPopMatrix();//Sale stack brazo 3.1

glScalef(2.5, 0.25, 0.25);

glRotatef(180, 1, 0, 0);

glColor3f(1.0, 0.0, 0.0);

prisma();

glPopMatrix();//Sale Brazo 3

glPushMatrix();//Brazo 4

glTranslatef(0.0f, 0.2f, 0.35f);

glRotatef(-90, 0.0, 1.0, 0.0);

glRotatef(movBrazo1, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix();//Brazo 4.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo11, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix(); //Brazo 4.1.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo111, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glScalef(2.5, 0.25, 0.25);

glColor3f(1.0, 0.2, 0.0);

prisma();

glPopMatrix(); // Sale del stack del Brazo 4.1.1

glScalef(2.5, .25, 0.25);

glColor3f(1.0, 1.0, 0.0);

prisma();

glPopMatrix();//Sale stack brazo 4.1

glScalef(2.5, 0.25, 0.25);

glRotatef(180, 1, 0, 0);

glColor3f(1.0, 0.0, 0.0);

prisma();

glPopMatrix();//Sale Brazo 4

glPushMatrix();//Brazo 5

glTranslatef(0.35f, 0.2f, 0.35f);

glRotatef(-45, 0.0, 1.0, 0.0);

glRotatef(movBrazo1, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix();//Brazo 5.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo11, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix(); //Brazo 5.1.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo111, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glScalef(2.5, 0.25, 0.25);

glColor3f(1.0, 0.2, 0.0);

prisma();

glPopMatrix(); // Sale del stack del Brazo 5.1.1

glScalef(2.5, .25, 0.25);

glColor3f(1.0, 1.0, 0.0);

prisma();

glPopMatrix();//Sale stack brazo 5.1

glScalef(2.5, 0.25, 0.25);

glRotatef(180, 1, 0, 0);

glColor3f(1.0, 0.0, 0.0);

prisma();

glPopMatrix();//Sale Brazo 5

glPushMatrix();//Brazo 6

glTranslatef(0.35f, 0.2f, -0.35f);

glRotatef(45, 0.0, 1.0, 0.0);

glRotatef(movBrazo1, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix();//Brazo 6.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo11, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix(); //Brazo 6.1.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo111, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glScalef(2.5, 0.25, 0.25);

glColor3f(1.0, 0.2, 0.0);

prisma();

glPopMatrix(); // Sale del stack del Brazo 6.1.1

glScalef(2.5, .25, 0.25);

glColor3f(1.0, 1.0, 0.0);

prisma();

glPopMatrix();//Sale stack brazo 6.1

glScalef(2.5, 0.25, 0.25);

glRotatef(180, 1, 0, 0);

glColor3f(1.0, 0.0, 0.0);

prisma();

glPopMatrix();//Sale Brazo6

glPushMatrix();//Brazo 7

glTranslatef(-0.35f, 0.2f, 0.35f);

glRotatef(225, 0.0, 1.0, 0.0);

glRotatef(movBrazo1, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix();//Brazo 7.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo11, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix(); //Brazo 7.1.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo111, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glScalef(2.5, 0.25, 0.25);

glColor3f(1.0, 0.2, 0.0);

prisma();

glPopMatrix(); // Sale del stack del Brazo 7.1.1

glScalef(2.5, .25, 0.25);

glColor3f(1.0, 1.0, 0.0);

prisma();

glPopMatrix();//Sale stack brazo 7.1

glScalef(2.5, 0.25, 0.25);

glRotatef(180, 1, 0, 0);

glColor3f(1.0, 0.0, 0.0);

prisma();

glPopMatrix();//Sale Brazo7

glPushMatrix();//Brazo 8

glTranslatef(-0.35f, 0.2f, -0.35f);

glRotatef(135, 0.0, 1.0, 0.0);

glRotatef(movBrazo1, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix();//Brazo 8.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo11, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glPushMatrix(); //Brazo 8.1.1

glTranslatef(1.25f, 0.0f, 0.0f);

glRotatef(movBrazo111, 0.0, 0.0, 1.0);

glTranslatef(1.25f, 0.0f, 0.0f);

glScalef(2.5, 0.25, 0.25);

glColor3f(1.0, 0.2, 0.0);

prisma();

glPopMatrix(); // Sale del stack del Brazo 8.1.1

glScalef(2.5, .25, 0.25);

glColor3f(1.0, 1.0, 0.0);

prisma();

glPopMatrix();//Sale stack brazo 8.1

glScalef(2.5, 0.25, 0.25);

glRotatef(180, 1, 0, 0);

glColor3f(1.0, 0.0, 0.0);

prisma();

glPopMatrix();//Sale Brazo8

glScalef(0.7, 1.0, 0.7);

glRotatef(90, 1, 0, 0);

glColor3f(1.0, 1.0, 0.0);

prisma();

glPopMatrix();//sale stack punta brazos

}

Punto 5:

Se crearon 2 funciones para los diferentes modelos 3ds que luego serían dibujados donde se deseara.

void Dragon()

{

//glPushMatrix();

// glDisable(GL\_COLOR\_MATERIAL);

glColor3f(0.1, 0.8, 0.1);

glRotatef(10, 1, 0, 0);

glScalef(0.01, 0.01, 0.01);

dragon.GLrender(NULL, \_SHADED, 1.0);

//glPopMatrix();

}

void Craneo()

{

//glPushMatrix();

//glDisable(GL\_COLOR\_MATERIAL);

glColor3f(0.9, 0.9, 0.9);

glTranslatef(1.5, 1.6, 3.7);

glScalef(0.3, 0.3, 0.3);

glRotatef(-10, 1, 0, 0);

craneo.GLrender(NULL, \_SHADED, 1.0);

//glPopMatrix();

}

//Obstáculo 3ds

glPushMatrix();

Craneo();//Bum1

glPopMatrix();

glPushMatrix();

glTranslatef(-4, -.3, 2);

Craneo();//Bum2

glPopMatrix();

glPushMatrix();

glTranslatef(0, -.7, 4);

Craneo();//Bum3

glPopMatrix();

//Obstaculo 3ds

glPushMatrix();//Dragon1 arriba

glTranslatef(-4.8, 0.5, 10);

Dragon();

glPopMatrix();

glPushMatrix();//Dragon2 abajo

glTranslatef(-6, 1.75, 2);

Dragon();

glPopMatrix();

Punto 6:

Canica 1 detona su animación con el click del mouse.

//Canica1

glPushMatrix();

glColor3f(0.30754, 0.30754, 0.30754);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, EarthDiffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, EarthSpecular);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, EarthAmbient);

glMaterialfv(GL\_FRONT, GL\_SHININESS, EarthShininess);

glTranslatef(xPos1, yPos1, zPos1);

glRotatef(10, 1, 0, 0);

glRotatef(giro\_Bol, -1, 0, 0);

canica1.esfera(0.25, 30, 30, 0);

glPopMatrix();

//Animación canica 1

if (g\_fanimacion)

{

if (adelante)

{

if (zPos1 >= 1.20) {

giro\_Bol += 10;

zPos1 -= 1;

yPos1 += 0.14;

}

else if (zPos1 > 0) {

giro\_Bol += 10;

zPos1 -= .2;

xPos1 -= .2;

yPos1 += 0.0;

}

}

else

{

if (zPos1 < 4) {

giro\_Bol -= 10;

zPos1 += .25;

yPos1 -= 0.045;

xPos1 -= 0.25;

}

else if (zPos1 < 7) {

giro\_Bol -= 10;

zPos1 += 0.25;

yPos1 -= 0.045;

xPos1 += 0.2;

}

else if (zPos1 < 9) {

giro\_Bol -= 10;

zPos1 += 0.25;

yPos1 -= 0.03;

xPos1 -= 1.0;

}

else if (zPos1 < 12) {

giro\_Bol -= 10;

zPos1 += 0.375;

yPos1 -= 0.045;

xPos1 += 0.5;

}

else if (zPos1 < 13.19) {

giro\_Bol -= 10;

zPos1 += .125;

yPos1 -= 0.025;

xPos1 -= 0.15;

printf("\n %f \n" ,zPos1);

}

else if (zPos1 < 13.40) {

zPos1 =13.50;

yPos1 = -0.25;

xPos1 = 7.0;

}

}

if (zPos1 <= 0.0)

adelante = false;

if (zPos1 >= 13.50)

adelante = true;

}

else

{

zPos1 =12.20;

yPos1 = -0.25;

xPos1 = 7.0;

}

Punto 7:

Canica 2 con animación por keyframes.

//Canica2

glPushMatrix();

glColor3f(0.30754, 0.30754, 0.30754);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, EarthDiffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, EarthSpecular);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, EarthAmbient);

glMaterialfv(GL\_FRONT, GL\_SHININESS, EarthShininess);

glTranslatef(posX, posY, posZ);

glRotatef(10, 1, 0, 0);

glRotatef(giro\_Bol, -1, 0, 0);

canica2.esfera(0.25, 30, 30, 0);

glPopMatrix();

//Movimiento por keyframes

if (play)

{

if (i\_curr\_steps >= i\_max\_steps) //end of animation between frames?

{

playIndex++;

if (playIndex > FrameIndex - 2) //end of total animation?

{

printf("Fin de animacion por keyframes\n");

playIndex = 0;

play = false;

}

else //Next frame interpolations

{

printf("Play Index: %d\n", playIndex);

printf("\tCoordenadas ( %0.2f, %0.2.f, %0.2.f)\n", posX, posY, posZ);

i\_curr\_steps = 0; //Reset counter

//Interpolation

KeyFrame[playIndex].incX = (KeyFrame[playIndex + 1].posX - KeyFrame[playIndex].posX) / i\_max\_steps; //100 frames

KeyFrame[playIndex].incY = (KeyFrame[playIndex + 1].posY - KeyFrame[playIndex].posY) / i\_max\_steps; //100 frames

KeyFrame[playIndex].incZ = (KeyFrame[playIndex + 1].posZ - KeyFrame[playIndex].posZ) / i\_max\_steps;

KeyFrame[playIndex].DerFlipI = (KeyFrame[playIndex + 1].DerFlip - KeyFrame[playIndex].DerFlip) / i\_max\_steps;

}

}

else

{ //Draw information

posX += KeyFrame[playIndex].incX;

posY += KeyFrame[playIndex].incY;

posZ += KeyFrame[playIndex].incZ;

DerFlip += KeyFrame[playIndex].DerFlipI;

i\_curr\_steps++;

}

}

Punto 8:

 Vista de tercera persona de personaje jerárquico que recorre la mesa por dentro.

//Personaje

glPushMatrix();

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, CirculoDiffuse);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, CirculoSpecular);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, CirculoAmbient);

glMaterialfv(GL\_FRONT, GL\_SHININESS, CirculoShininess);

glTranslatef(xa, ya, z);

glRotatef(10, 1, 0, 0);

glRotatef(180, 0, 1, 0);

glScalef(0.5, 0.5, 0.5);

Morty();

glPopMatrix();

void Morty()

{

glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_SPECULAR, whiteSpecularMaterial);

glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_SHININESS, mShininess);

glPushMatrix();//Pecho

glScalef(0.5, 0.5, 0.5);

fig7.cilindro(1, 2, 24, amarillo.GLindex);

glPushMatrix();//Cuello

glTranslatef(0, 2.0, 0.0);

fig7.cilindro(0.25, 0.5, 15, carne.GLindex);

glPushMatrix();//Cabeza

glTranslatef(0, 1.4, 0);

fig7.esfera(1.2, 24, 24, carne.GLindex);

glPushMatrix();//Cabello

glTranslatef(0, 0.3, -0.3);

fig7.esfera(1.1, 24, 24, cafe.GLindex);

glPopMatrix();

glPopMatrix();

glPopMatrix();

glPushMatrix(); //Brazo derecho-->

glTranslatef(1.25, 1.5, 0);

glRotatef(rotHombroDer, 0.0, 1.0, 0.0);

fig7.esfera(0.5, 12, 12, amarillo.GLindex);

glPushMatrix();

glTranslatef(0, -2, 0);

glRotatef(0, 0.0, 0.0, 1.0);

fig7.cilindro(0.5, 2, 24, amarillo.GLindex);

glPushMatrix();

glTranslatef(0, -0.3, 0);

fig7.esfera(0.4, 12, 12, carne.GLindex);

glPopMatrix();

glPopMatrix();

glPopMatrix();

glPushMatrix(); //Brazo izquierdo-->

glTranslatef(-1.25, 1.5, 0);

glRotatef(rotHombroDer, 0.0, 1.0, 0.0);

fig7.esfera(0.5, 12, 12, amarillo.GLindex);

glPushMatrix();

glTranslatef(0, -2, 0);

glRotatef(0, 0.0, 0.0, 1.0);

fig7.cilindro(0.5, 2, 24, amarillo.GLindex);

glPushMatrix();

glTranslatef(0, -0.3, 0);

fig7.esfera(0.4, 12, 12, carne.GLindex);

glPopMatrix();

glPopMatrix();

glPopMatrix();

glPushMatrix();//Cintura

glTranslatef(0, -1.5, 0);

glPushMatrix(); //Pie Derecho -->

glTranslatef(0.7, -0.75, 0);

fig7.cilindro(0.5, 2.3, 16, azul.GLindex);

glPushMatrix();

glTranslatef(0, -0.5, 0.3);

fig7.prisma1(0.2, 1.0, 1.5, cafe.GLindex, cafe.GLindex, cafe.GLindex);

glPopMatrix();

glPopMatrix();

glPushMatrix(); //Pie Izquierdo -->

glTranslatef(-0.7, -0.75, 0);

fig7.cilindro(0.5, 2.3, 16, azul.GLindex);

glPushMatrix();

glTranslatef(0, -0.5, 0.3);

fig7.prisma1(0.2, 1.0, 1.5, cafe.GLindex, cafe.GLindex, cafe.GLindex);

glPopMatrix();

glPopMatrix();

glPopMatrix();

glPopMatrix();

}

Punto 9:

Cambio de cámaras con memoria de posición.

case 'c': //cambiamos camara

case 'C':

switchCam = !switchCam;

break;

//--------------------------------------

if (switchCam) {

glRotatef(g\_lookupdown,1.0f,0,0);

gluLookAt( objCamera.mPos.x, objCamera.mPos.y, objCamera.mPos.z,

objCamera.mView.x, objCamera.mView.y, objCamera.mView.z,

objCamera.mUp.x, objCamera.mUp.y, objCamera.mUp.z);

}

else{

glRotatef(g\_lookupdown1, 1.0f, 0, 0);

gluLookAt(objCamera1.mPos.x, objCamera1.mPos.y, objCamera1.mPos.z,

objCamera1.mView.x, objCamera1.mView.y, objCamera1.mView.z,

objCamera1.mUp.x, objCamera1.mUp.y, objCamera1.mUp.z);

}

Punto 10:

Se añadió la textura al fondo del tablero y 3 texturas con animación.

void AdornosInteriores() {

//Textura luz1

glPushMatrix();

glTranslatef(5, 1, 7);

glRotatef(-90, 1, 0, 0);

glScalef(4.0, 4.0, 1);

fig\_baners.TexturaConCorrimiento(luz1.GLindex, var, 0);

glPopMatrix();

//Textura luz2 Lazer verde

glPushMatrix();

glTranslatef(-0.2, 0.0, 11.5);

glRotatef(100, 1, 0, 0);

glScalef(1.0, 4.5, 1);

fig\_baners.TexturaConCorrimiento(luz2.GLindex, 0, var);

glPopMatrix();

//Textura luz3

glPushMatrix();

glTranslatef(4, 1.8, 1.3);

glRotatef(180, 0, 1, 0);

glRotatef( 80, 1, 0, 0);

glScalef(3.0, 1.0, 1);

fig\_baners.TexturaConCorrimiento(luz1.GLindex, var, 0);

glPopMatrix();

}

Punto 11:

Modelo jerárquico para el tablero hecho previamente.

void TableroJerarquico(GLuint text1)

{

glPushMatrix();

glTranslatef(0.0, 1.5, 7.5);

glPushMatrix();

glTranslatef(-7.0, -2.7, -7.8);

glRotatef(10, 1, 0, 0);

glPushMatrix();

glTranslatef(14.0, 0.0, 0.0);

glScalef(1.0, 5, 1.0);

fig\_mesa.pata(madera.GLindex);//Pata Derecha Trasera

glPopMatrix();

glScalef(1.0, 5, 1.0);

fig\_mesa.pata(madera.GLindex);//Pata Izquierda Trasera

glPopMatrix();

glPushMatrix();

glTranslatef(-7.0, -3.5, 7.3);

glPushMatrix();

glTranslatef(14, 0.0, 0);

glScalef(1.0, 3.0, 1.0);

fig\_mesa.pata(madera.GLindex); //Pata Delantera Derecha

glPopMatrix();

glScalef(1.0, 3.0, 1.0);

fig\_mesa.pata(madera.GLindex);//Pata Delantera Izquerda

glPopMatrix();

glRotatef(10, 1, 0, 0);

glScalef(15.0, 3.0, 16.0);

fig\_mesa.caja(madera.GLindex,vidrio.GLindex); //Sale Mesa

glPopMatrix();

}

## Resultado final



# Sección Manual de usuario:

**‘Tecla’ –** Función

**Animaciones**

**‘mouseClickIzquierdo’ –** Activa/desactiva la animación de la canica #1 (recorrido predeterminado), y activa la animación la palanca.

**‘p’/’P’ –** Activa/desactiva la animación de la canica #2 hecha por keyframes.

**Camara 1 (Vista General)**

**‘c’/’C’ –** Cambia entre la camara#1 y la camara#2.

**‘w’/’W’ –** Mueve la camara#1 (vista general) en profundidad (eje z**–**).

**‘s’/’S’ –** Mueve la camara#1 (vista general) alejandose (eje z**+**).

**‘a’/’A’ –** Mueve la camara#1 (vista general) hacia la izquierda (eje x**–**).

**‘d’/’D’ –** Mueve la camara#1 (vista general) hacia la derecha (eje x+).

**‘7’ –** Mueve la camara#1 (vista general) hacia arriba (eje y+).

**‘9’ –** Mueve la camara#1 (vista general) hacia abajo (eje y**–**).

**‘GLUT\_KEY\_PAGE\_UP’ –** Mueve la camara#1 (vista general) hacia arriba (eje y+).

**‘GLUT\_KEY\_PAGE\_DOWN’ –** Mueve la camara#1 (vista general) hacia abajo (eje y**–**).

**‘GLUT\_KEY\_UP’ –** Gira la camara#1 (vista general) hacia arriba.

**‘GLUT\_KEY\_DOWN’ –** Gira la camara#1 (vista general) hacia abajo.

**‘GLUT\_KEY\_LEFT’ –** Gira la camara#1 (vista general) hacia la izquierda.

**‘GLUT\_KEY\_RIGHT’ –** Gira la camara#1 (vista general) hacia la derecha.

**Camara 2 (Vista Canica#3)**

**Igual que cámara 1.**

**KEYFRAMES**

**‘r’/’R’ –** Guarda un cuadro de Keyframe.

**‘p’/’P’ –** Reproduce todas los cuadros guardados de Keyframe.

**Luces**

**‘j’/’J’ –** Apaga/Prende la luz del tablero.

**‘k’/’K’ –** Apaga/Prende la sirena en la esquina superior izquierda del tablero.

**‘l’/’L’ –** Apaga/Prende una luz al centro del tablero de color amarillo.

**Otros**

**‘Esc’ –** Termina la ejecucion del programa.