#define \_USE\_MATH\_DEFINES

# define M\_PI 3.14159265358979323846 /\* pi \*/

#include <cstdlib>

#include <ctime>

#include <iostream>

#include "GL/glut.h"

#include <math.h>

double random() {

return ((double)rand() / (RAND\_MAX));

}

double randomx() {

return (rand() % 390) + 150;

}

double randomy() {

return (rand() % 100) + 140;

}

int \_h = 0, \_m = 0, \_n = 0, \_c = 0, \_eex = 20, \_eey = 20, \_eez = 20, \_ecx = 20, \_ecy = 20, \_ecz = 50, \_ec = 20, \_botea = 1, \_boteb = 1, \_botec = 1, \_boted = 1, \_botee = 1, \_botef = 1, \_xe = 0, \_xc = 0, \_xl = 0, \_re = 0, \_rc = 0, \_rl = 0;

int \_bbotea = 1, \_bboteb = 1, \_bbotec = 1, \_bboted = 1, \_bbbotea = 1, \_bbboteb = 1, \_bbbotec = 1, \_bbboted = 1;

double \_te = 1, \_tc = 1, \_tcc = 1, \_ccr = 1, \_ccg = 0, \_ccb = 0, \_cer = 0, \_ceg = 1, \_ceb = 0, \_cccr = 0, \_cccg = 0, \_cccb = 1, r = 0;

double i = 5, j = 5, k = 5, ii = 5, jj = 5, kk = 5, iii = 5, jjj = 5, kkk = 5, rrx = randomx(), rrry = randomy(), rrxx = randomx(), rrryy = randomy(), rrxxx = randomx(), rrryyy = randomy();

double minbbx = 130, minbby = 120, minbbz = 1, maxbbx = 560, maxbby = 260, maxbbz = -260;

void init(void) {

glClearColor(1.0, 1.0, 1.0, 0.0);

glMatrixMode(GL\_MODELVIEW);

glOrtho(0.0, 600.0, 0.0, 300.0, 0.0, 300.0);//Esto es lo que afecta verdaderamente a la ventana

}

bool collisionDetection(double px, double py, double pz) {

if ((px >= minbbx && px <= maxbbx) && (py >= minbby && py <= maxbby) && (pz >= minbbz && pz <= maxbbz)) {

return false;

}

else {

return true;

//PrevPos = Pos

}

}

void dibujaEsfera(double px, double py, double pz) {

glEnable(GL\_BLEND);

glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);

if (collisionDetection(px, py, pz)) {

glColor4f(\_ccr, \_ccg, \_ccb, \_tc); // Red ball

glPushMatrix();

glTranslatef(px, py, pz);

if (\_re != 0) {

glRotated(\_re, 1.0, 0.0, 0.0);

glRotated(\_re, 0.0, 1.0, 0.0);

glRotated(\_re, 0.0, 0.0, 1.0);

\_re = \_re + 1;

}

glutSolidSphere(\_ecx, \_ecy, \_ecz);

glPopMatrix();

}

glDisable(GL\_BLEND);

}

void dibujaElipsoide(double px, double py, double pz) {

glEnable(GL\_BLEND);

glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);

if (collisionDetection(px, py, pz)) {

glColor4f(\_cer, \_ceg, \_ceb, \_te);

glPushMatrix();

glTranslatef(px, py, pz);

glScalef(1.2, 0.7, 0.8);

if (\_rl != 0) {

glRotated(\_rl, 1.0, 0.0, 0.0);

glRotated(\_rl, 0.0, 1.0, 0.0);

glRotated(\_rl, 0.0, 0.0, 1.0);

\_rl = \_rl + 1;

}

glutSolidSphere(\_eex, \_eey, \_eez);

glPopMatrix();

}

glDisable(GL\_BLEND);

}

void dibujaCubo(double px, double py, double pz) {

glEnable(GL\_BLEND);

glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);

if (collisionDetection(px, py, pz)) {

glColor4f(\_cccr, \_cccg, \_cccb, \_tcc);

glPushMatrix();

glTranslatef(px, py, pz);

if (\_rc != 0) {

glRotated(\_rc, 1.0, 0.0, 0.0);

glRotated(\_rc, 0.0, 1.0, 0.0);

glRotated(\_rc, 0.0, 0.0, 1.0);

\_rc = \_rc + 1;

}

glutSolidCube(\_ec);

glPopMatrix();

}

glDisable(GL\_BLEND);

}

void RenderString(float x, float y, void \*font, const char \*string) {

const char \*c;

glRasterPos2f(x, y);

for (c = string; \*c != '\0'; c++) {

glutBitmapCharacter(font, \*c);

}

}

void interfaz() {

int j = 0, pos = 40;

float angle = 0;

//glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0, 0, 0);

glBegin(GL\_QUADS); //Rectángulos

//Marco

//Izquierdo

glVertex2i(4, 4);

glVertex2i(4, 296);

glVertex2i(8, 296);

glVertex2i(8, 4);

//Superior

glVertex2i(4, 296);

glVertex2i(596, 296);

glVertex2i(596, 292);

glVertex2i(4, 292);

//Derecho

glVertex2i(596, 296);

glVertex2i(592, 296);

glVertex2i(592, 4);

glVertex2i(596, 4);

//Inferior

glVertex2i(596, 4);

glVertex2i(596, 8);

glVertex2i(4, 8);

glVertex2i(4, 4);

//Marco de comandos

//Izquierdo

glVertex2i(20, 20);

glVertex2i(20, 80);

glVertex2i(22, 80);

glVertex2i(22, 20);

//Superior

glVertex2i(20, 80);

glVertex2i(580, 80);

glVertex2i(580, 78);

glVertex2i(20, 78);

//Derecho

glVertex2i(580, 80);

glVertex2i(580, 20);

glVertex2i(578, 20);

glVertex2i(578, 80);

//Inferior

glVertex2i(580, 20);

glVertex2i(20, 20);

glVertex2i(20, 22);

glVertex2i(580, 22);

glEnd();

//Texto

void \* font = GLUT\_BITMAP\_HELVETICA\_18;

RenderString(25, 65, font, "E");

RenderString(40, 65, font, "Escalar");

RenderString(25, 50, font, "M");

RenderString(40, 50, font, "Mover");

RenderString(25, 35, font, "C");

RenderString(40, 35, font, "Cambiar de Color");

RenderString(390, 65, font, "R");

RenderString(415, 65, font, "Rotar");

RenderString(390, 50, font, "T");

RenderString(415, 50, font, "Transparente");

RenderString(390, 35, font, "ESC");

RenderString(415, 35, font, "Salir");

//Objetos de selección

//Esfera

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_POLYGON);

for (j = 0; j < 360; j++) {//Circulo superior izquierdo grande

angle = j \* 3.142 / 180;

//El izquierdo es x y el derecho es y; cualquier modificación sobre pos afecta la posición, sobre angle afecta la forma.

glVertex2f((pos + 20) + 30 \* cos(angle), (pos + 200) + 30 \* sin(angle));

}

glEnd();

//Cuadrado

glColor3f(0.0, 0.0, 1.0);

glBegin(GL\_QUADS);

glVertex3f(35, 150, 0);

glVertex3f(35, 200, 0);

glVertex3f(85, 200, 0);

glVertex3f(85, 150, 0);

glEnd();

//Elipse

glColor3f(0.0, 1.0, 0.0);

glBegin(GL\_POLYGON);

for (j = 0; j < 360; j++) {//Circulo superior izquierdo grande

angle = j \* 3.142 / 180;

//El izquierdo es x y el derecho es y; cualquier modificación sobre pos afecta la posición, sobre angle afecta la forma.

glVertex2f((pos + 20) + 40 \* cos(angle), (pos + 80) + 20 \* sin(angle));

}

glEnd();

glPushMatrix();

}

void cuadroVisualizacion() {

//glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 0.0, 0.0);

glBegin(GL\_QUADS);

//Rectangulos de visualización

for (int i = 0; i <= 455; i += 5) {

glVertex3i(115 + i, 105, -3);

glVertex3i(115 + i, 275, 0);

glVertex3i(120 + i, 275, 0);

glVertex3i(120 + i, 105, -3);

i += 5;

}

//Marco de visualización

//Izquierdo

glVertex2i(110, 100);

glVertex2i(112, 100);

glVertex2i(112, 280);

glVertex2i(110, 280);

glVertex3f(110, 100, 0);

glVertex3f(110, 100, -280);

glVertex3f(110, 280, -280);

glVertex3f(110, 280, 0);

//Superior

glVertex2i(110, 280);

glVertex2i(580, 280);

glVertex2i(580, 278);

glVertex2i(110, 278);

glVertex3f(110, 280, 0);

glVertex3f(110, 280, -280);

glVertex3f(580, 280, -280);

glVertex3f(580, 280, 0);

//Derecho

glVertex2i(580, 280);

glVertex2i(580, 100);

glVertex2i(578, 100);

glVertex2i(578, 280);

glVertex3f(580, 280, 0);

glVertex3f(580, 280, -280);

glVertex3f(580, 100, -280);

glVertex3f(580, 100, 0);

//Inferior

glVertex2i(580, 100);

glVertex2i(110, 100);

glVertex2i(110, 102);

glVertex2i(580, 102);

glVertex3f(580, 100, 0);

glVertex3f(580, 100, -280);

glVertex3f(110, 100, -280);

glVertex3f(110, 100, 0);

glEnd();

//Cuadros invisibles

glColorMask(GL\_FALSE, GL\_FALSE, GL\_FALSE, GL\_FALSE);

glBegin(GL\_QUADS);

//Enfrente

glVertex3f(110, 100, 0);

glVertex3f(110, 280, 0);

glVertex3f(580, 280, 0);

glVertex3f(580, 100, 0);

//Detras

glVertex3f(110, 100, -280);

glVertex3f(110, 280, -280);

glVertex3f(580, 280, -280);

glVertex3f(580, 100, -280);

glEnd();

glColorMask(GL\_TRUE, GL\_TRUE, GL\_TRUE, GL\_TRUE);

}

void escena() {

glClear(GL\_COLOR\_BUFFER\_BIT);

interfaz();

cuadroVisualizacion();

if (\_m != 0) {

if (\_xe == 1) {

if (rrry + j == 250) {

\_botea = -1;

\_boteb = 1;

}

if (rrry + j == 130) {

\_botea = 1;

\_boteb = -1;

}

if (rrx + i == 560) {

\_botec = -1;

\_boted = 1;

}

if (rrx + i == 135) {

\_botec = 1;

\_boted = -1;

}

if (\_botea == 1) {

j = j + 1;

}

if (\_boteb == 1 && \_botea == -1) {

j = j - 1;

}

if (\_botec == 1) {

i = i + 1;

}

if (\_botec == -1 && \_boted == 1) {

i = i - 1;

}

if (k == 0) {

\_botee = 1;

\_botef = -1;

}

if (k == -20) {

\_botef = 1;

\_botee = -1;

}

if (\_botee == 1) {

k = k - 1;

}

if (\_botee == -1 && \_botef == 1) {

k = k + 1;

}

}

dibujaEsfera(rrx + i, rrry + j, 0 - k);

}

if (\_c != 0) {

if (\_xc == 1) {

if (rrryy + jj == 250) {

\_bbotea = -1;

\_bboteb = 1;

}

if (rrryy + jj == 130) {

\_bbotea = 1;

\_bboteb = -1;

}

if (rrxx + ii == 560) {

\_bbotec = -1;

\_bboted = 1;

}

if (rrxx + ii == 135) {

\_bbotec = 1;

\_bboted = -1;

}

if (\_bbotea == 1) {

jj = jj + 1;

}

if (\_bboteb == 1 && \_bbotea == -1) {

jj = jj - 1;

}

if (\_bbotec == 1) {

ii = ii + 1;

}

if (\_bbotec == -1 && \_bboted == 1) {

ii = ii - 1;

}

}

dibujaCubo(rrxx + ii, rrryy + jj, 0);

}

if (\_n != 0) {

if (\_xl == 1) {

if (rrryyy + jjj == 250) {

\_bbbotea = -1;

\_bbboteb = 1;

}

if (rrryyy + jjj == 135) {

\_bbbotea = 1;

\_bbboteb = -1;

}

if (kkk == -20) {

\_bbbotec = -1;

\_bbboted = 1;

}

if (kkk == 0) {

\_bbbotec = 1;

\_bbboted = -1;

}

if (\_bbbotea == 1) {

jjj = jjj + 1;

}

if (\_bbboteb == 1 && \_bbbotea == -1) {

jjj = jjj - 1;

}

if (\_bbbotec == 1) {

kkk = kkk - 1;

}

if (\_bbbotec == -1 && \_bbboted == 1) {

kkk = kkk + 1;

}

}

dibujaElipsoide(rrxx, rrryyy + jjj, 0 - kkk);

}

glutSwapBuffers();

}

void keyboardDown(unsigned char key, int x, int y) {

if (\_h != 0) {

if (\_h == 1) {

switch (key) {

case '1':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ecx = \_ecx \* 1.1;

\_ecy = \_ecy \* 1.1;

\_ecz = \_ecz \* 1.1;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_ec = \_ec \* 1.1;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_eex = \_eex \* 1.1;

\_eey = \_eey \* 1.1;

\_eez = \_eez \* 1.1;

}

break;

case '2':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ecx = \_ecx \* 1.15;

\_ecy = \_ecy \* 1.15;

\_ecz = \_ecz \* 1.15;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_eex = \_eex \* 1.15;

\_eey = \_eey \* 1.15;

\_eez = \_eez \* 1.15;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_ec = \_ec \* 1.15;

}

break;

case '3':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ecx = \_ecx \* 1.2;

\_ecy = \_ecy \* 1.2;

\_ecz = \_ecz \* 1.2;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_ec = \_ec \* 1.2;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_eex = \_eex \* 1.2;

\_eey = \_eey \* 1.2;

\_eez = \_eez \* 1.2;

}

break;

case '4':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ecx = \_ecx \* 1.25;

\_ecy = \_ecy \* 1.25;

\_ecz = \_ecz \* 1.25;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_ec = \_ec \* 1.25;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_eex = \_eex \* 1.25;

\_eey = \_eey \* 1.25;

\_eez = \_eez \* 1.25;

}

break;

case '5':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ecx = \_ecx \* 1.3;

\_ecy = \_ecy \* 1.3;

\_ecz = \_ecz \* 1.3;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_ec = \_ec \* 1.3;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_eex = \_eex \* 1.3;

\_eey = \_eey \* 1.3;

\_eez = \_eez \* 1.3;

}

break;

case '6':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ecx = \_ecx \* 1.35;

\_ecy = \_ecy \* 1.35;

\_ecz = \_ecz \* 1.35;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_ec = \_ec \* 1.35;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_eex = \_eex \* 1.35;

\_eey = \_eey \* 1.35;

\_eez = \_eez \* 1.35;

}

break;

case '7':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ecx = \_ecx \* 1.4;

\_ecy = \_ecy \* 1.4;

\_ecz = \_ecz \* 1.4;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_ec = \_ec \* 1.4;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_eex = \_eex \* 1.4;

\_eey = \_eey \* 1.4;

\_eez = \_eez \* 1.4;

}

break;

case '8':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ecx = \_ecx \* 1.45;

\_ecy = \_ecy \* 1.45;

\_ecz = \_ecz \* 1.45;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_ec = \_ec \* 1.45;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_eex = \_eex \* 1.45;

\_eey = \_eey \* 1.45;

\_eez = \_eez \* 1.45;

}

break;

case '9':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ecx = \_ecx \* 1.5;

\_ecy = \_ecy \* 1.5;

\_ecz = \_ecz \* 1.5;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_ec = \_ec \* 1.5;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_eex = \_eex \* 1.5;

\_eey = \_eey \* 1.5;

\_eez = \_eez \* 1.5;

}

break;

case '0':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ecx = \_ecx \* 0.5;

\_ecy = \_ecy \* 0.5;

\_ecz = \_ecz \* 0.5;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_ec = \_ec \* 0.5;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_eex = \_eex \* 0.5;

\_eey = \_eey \* 0.5;

\_eez = \_eez \* 0.5;

}

break;

}

}

else {

switch (key) {

case '1':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_tc = \_tc \* 1;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_tcc = \_tcc \* 1;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_te = \_te \* 1;

}

break;

case '2':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_tc = \_tc \* .2;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_tcc = \_tcc \* .2;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_te = \_te \* .2;

}

break;

case '3':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_tc = \_tc \* .3;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_tcc = \_tcc \* .3;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_te = \_te \* .3;

}

break;

case '4':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_tc = \_tc \* .4;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_tcc = \_tcc \* .4;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_te = \_te \* .4;

}

break;

case '5':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_tc = \_tc \* .5;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_tcc = \_tcc \* .5;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_te = \_te \* .5;

}

break;

case '6':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_tc = \_tc \* .6;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_tcc = \_tcc \* .6;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_te = \_te \* .6;

}

break;

case '7':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_tc = \_tc \* .7;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_tcc = \_tcc \* .7;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_te = \_te \* .7;

}

break;

case '8':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_tc = \_tc \* .8;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_tcc = \_tcc \* .8;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_te = \_te \* .8;

}

break;

case '9':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_tc = \_tc \* .9;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_tcc = \_tcc \* .9;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_te = \_te \* .9;

}

break;

case '0':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_tc = \_tc \* 0;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_tcc = \_tcc \* 0;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_te = \_te \* 0;

}

break;

}

}

\_h = 0;

}

else {

switch (key) {

case 27:

exit(0);

break;

case 'm':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_xe = 1;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_xc = 1;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_xl = 1;

}break;

case 'c':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_ccr = random();

\_ccg = random();

\_ccb = random();

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_cccr = random();

\_cccg = random();

\_cccb = random();

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_cer = random();

\_ceg = random();

\_ceb = random();

} break;

case 'r':

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {//Esfera

\_re = 1;

}

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {//Cuadrado

\_rc = 1;

}

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {//Elipse

\_rl = 1;

}break;

case 'e':

\_h = 1;

break;

case 't':

\_h = 2;

break;

case 'p':

glRotatef(1, 0, 10, 0);

break;

case 'o':

glRotatef(1, 10, 10, 0);

break;

case 'l':

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

break;

case 'k':

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

break;

case 'z':

\_m = 0; \_c = 0; \_n = 0; \_eex = 20; \_eey = 20; \_eez = 50; \_ecx = 20; \_ecx = 20; \_ecz = 20; \_ec = 20;

\_tc = 1; \_tcc = 1; \_te = 1; \_ccr = 1; \_ccg = 0; \_ccb = 0; \_cer = 0; \_ceg = 1; \_ceb = 0; \_cccr = 0; \_cccg = 0; \_cccb = 1;

rrx = randomx(); rrry = randomy(); rrxx = randomx(); rrryy = randomy(); rrxxx = randomx(); rrryyy = randomy();

i = 5; j = 5; k = 5; ii = 5; jj = 5; kk = 5; iii = 5; jjj = 5; kkk = 5; \_botea = 1; \_boteb = 1; \_botec = 1; \_boted = 1; \_botee = 1; \_botef = 1, \_xe = 0, \_xc = 0, \_xl = 0, \_re = 0, \_rc = 0, \_rl = 0;

glPopMatrix();

break;

case 'x':

glPopMatrix();

break;

}

}

}

void click(int button, int state, int x, int y) {//Las coordenadas son al doble de grande y el eje x.y empieza en la esquina superior izquierda

switch (button) {

case GLUT\_LEFT\_BUTTON:

if (state == GLUT\_UP) {

//Esfera

if (x <= 180 && x >= 60 && y <= 180 && y >= 60) {

//glutPostRedisplay();

\_m = 1;

}

//Cuadrado

if (x <= 170 && x >= 70 && y <= 300 && y >= 200) {

//glutPostRedisplay();

\_c = 1;

}

//Elipse

if (x <= 200 && x >= 40 && y <= 400 && y >= 320) {

//glutPostRedisplay();

\_n = 1;

}

}break;

}

}

void main(int argc, char\*\* argv) {

glEnable(GL\_BLEND);

glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DEPTH | GLUT\_DOUBLE | GLUT\_RGBA | GLUT\_ALPHA);

glutInitWindowPosition(200, 0);

glutInitWindowSize(1200, 600);

glutCreateWindow("Proyecto Final");

init();

glutDisplayFunc(interfaz);

glutIdleFunc(escena);

glutKeyboardFunc(keyboardDown);

glutMouseFunc(click);

glutMainLoop();

}