

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

LABORATORIO DE COMPUTACIÓN GRÁFICA

EFEECTO COMPTON

ALUMNOS: MÉNDEZ GRANADOS RUTH CRISTINA

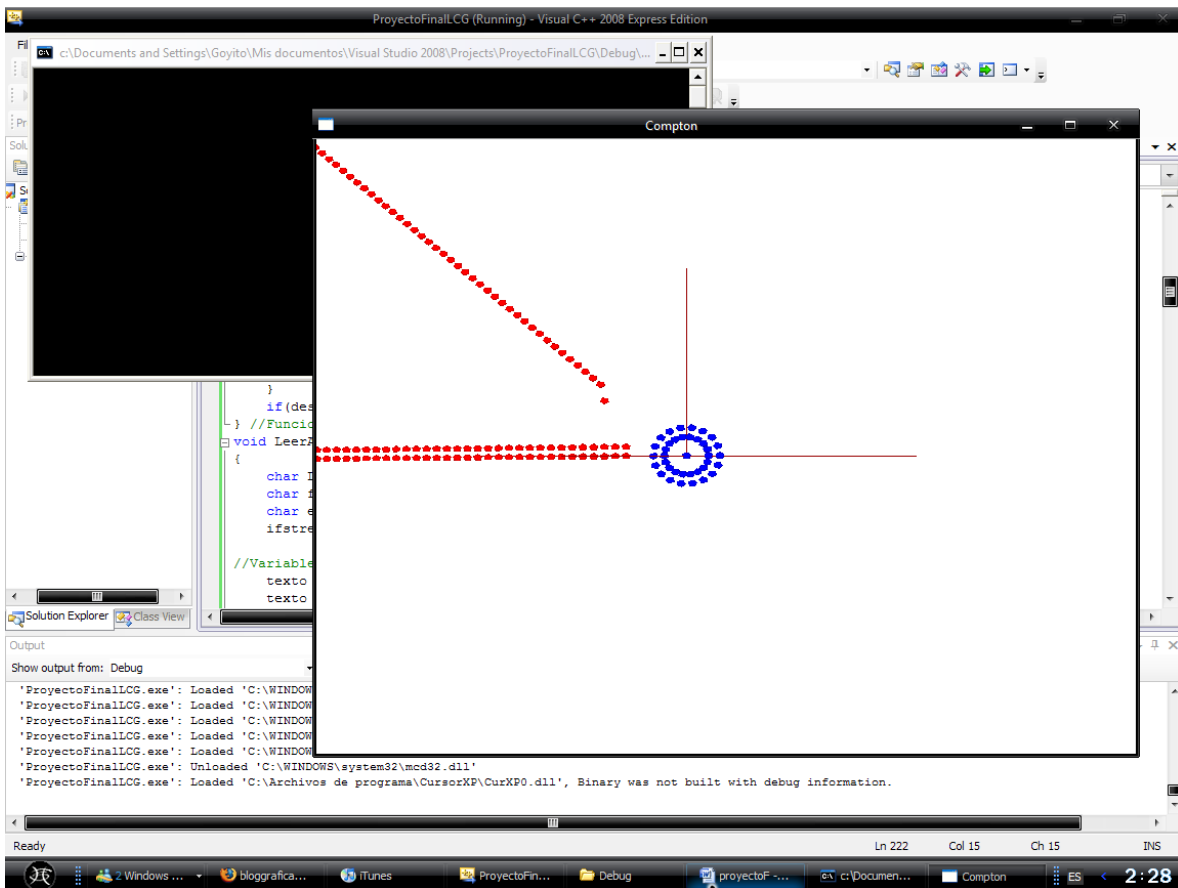
ROSALES VELA GREGORIO ALFONSO

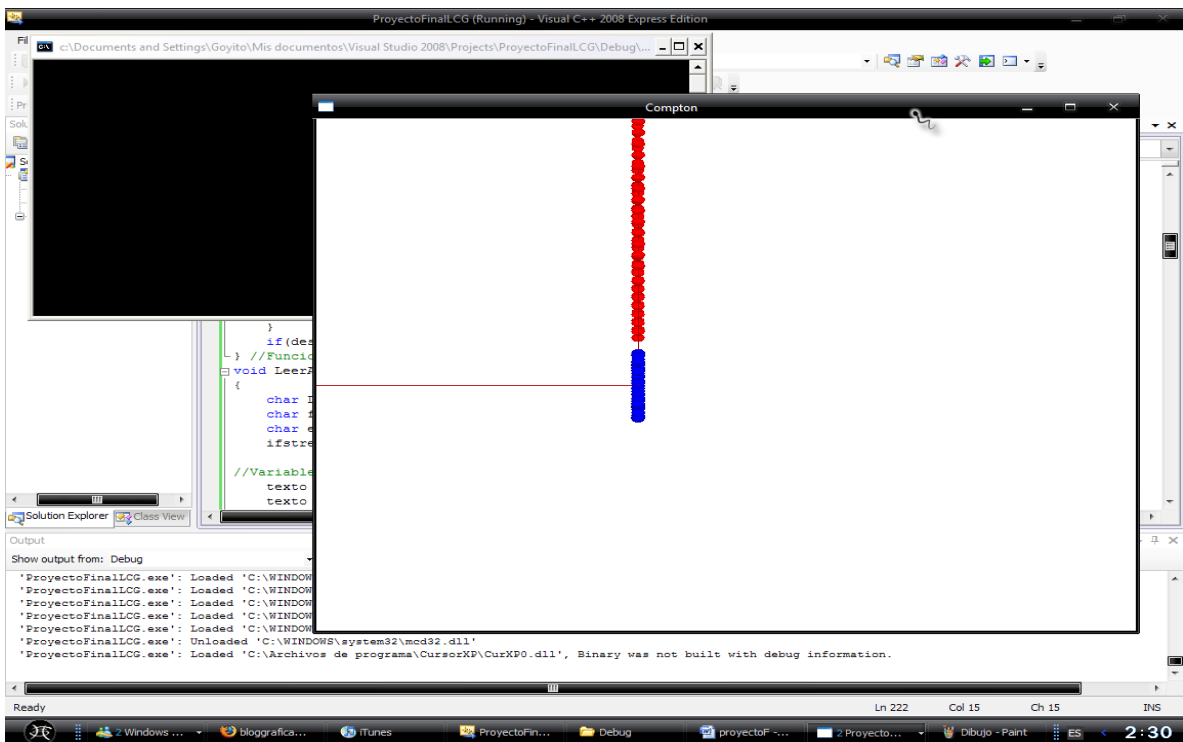
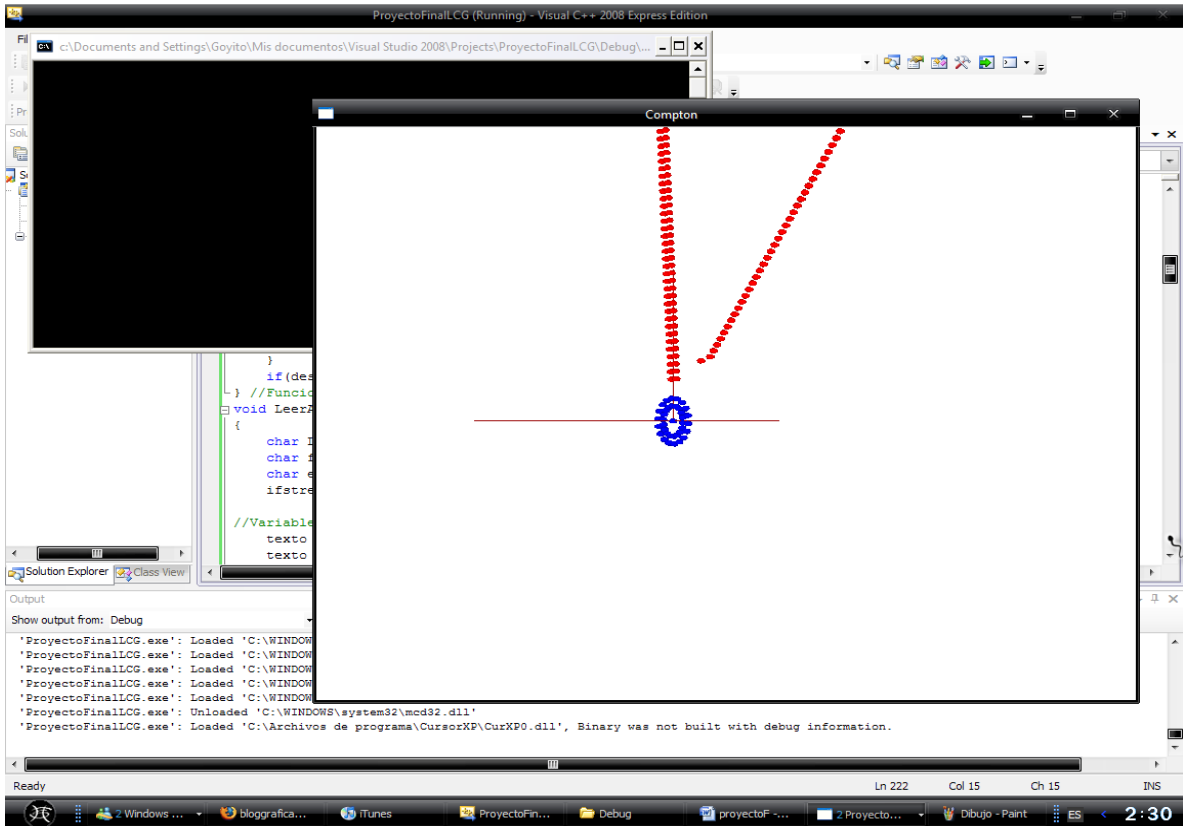
Objetivo:

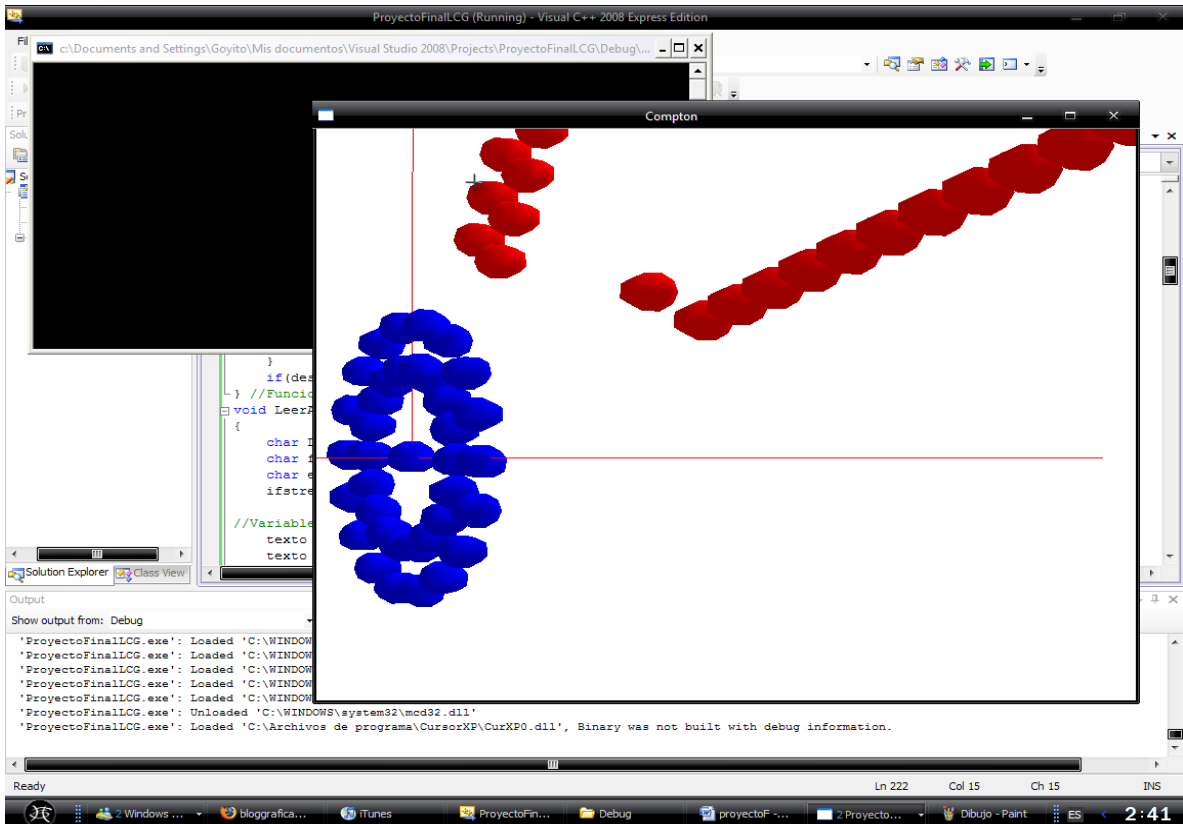
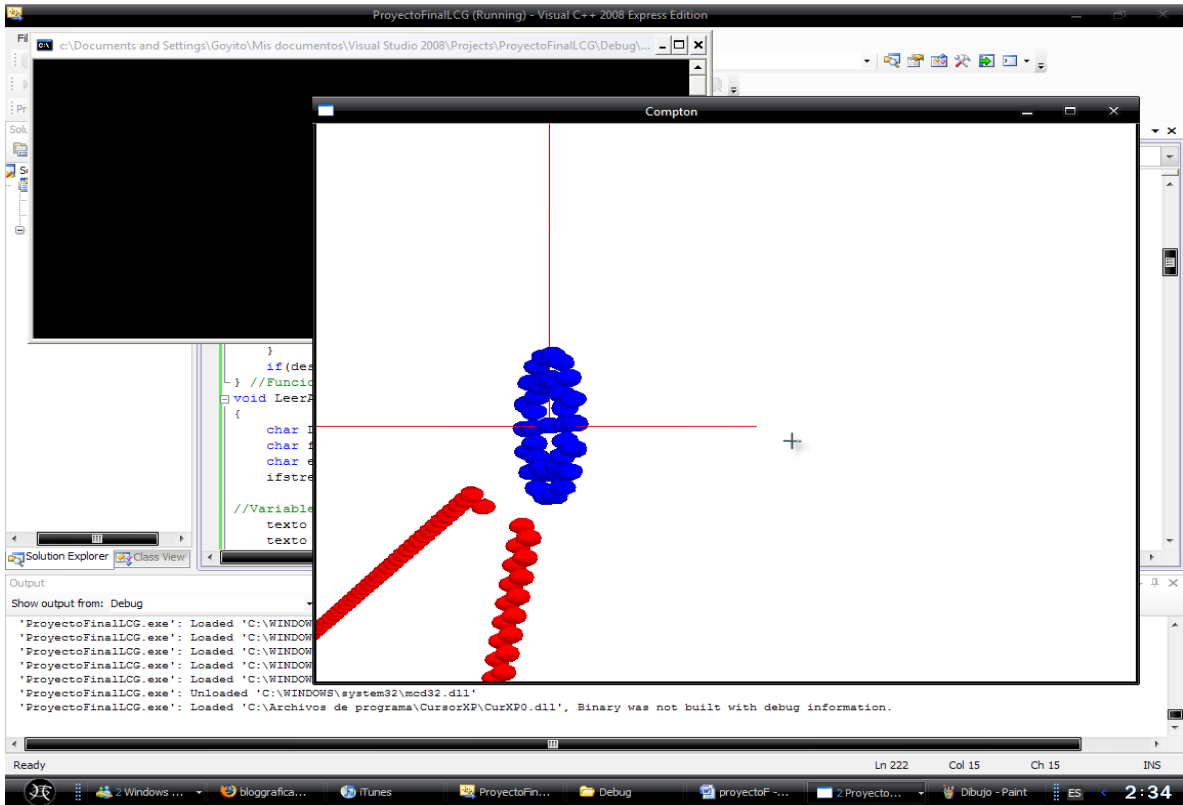
Que el alumno pueda implementar todos los conocimientos aprendidos durante el curso para poder realizar una animación que conlleve todo lo visto en el.

Introducción:

El efecto Compton es la desviación en movimiento del golpe de un fotón (rojo) contra un electrón (azul), donde ambos salen disparados en diferentes direcciones. visualmente se vera así: enseguida muestro el archivo datos.txt en el cual se realizó el proyecto, cabe mencionar que después de varias pruebas, pude comprobar que mientras mas grande era el valor de longitud de onda la velocidad del fotón era muy lenta. El orden de los datos en el archivo es el siguiente:







Codigo Fuente Utilizado:

```
#include "openglut.h"
#include <math.h>
#include <fstream>
#include <string>

using std::ifstream;
GLUquadricObj *quadobj, *discobj;
// variables del programa
long double h=6.63E-34;
long double w=0, E=0;
long double desf=0, dese=0;
long double LO;
long double foton;
long double electron;
long double Afoton=0, Aelectron=0, vangular=0, pos=0, pi=3.1416,
frecuencia=0.1, angle1=40, angle2=40;
GLuint fotones[10];

GLfloat light_Ambient [4] = { 0.4, 0.4, 0.4, 1.0};
GLfloat light_Diffuse [4] = { 0.7, 0.7, 0.7, 1.0};
GLfloat light_Position [4] = {20.0, 15.0, 10.0, 1.0};

GLfloat material [4] = {1.0, 0.2, 0.2, 1.0 };
GLfloat RedMaterial [4] = {1.0, 0.0, 0.0, 1.0 };
GLfloat GreenMaterial [4] = {0.0, 1.0, 0.0, 1.0 };
GLfloat BlueMaterial [4] = {0.0, 0.0, 1.0, 1.0 };
GLfloat WhiteMaterial [4] = {1.0, 1.0, 1.0, 1.0 };

void luces(void) {
    glEnable (GL_LIGHTING);
    glEnable (GL_LIGHT0);
    glLightfv(GL_LIGHT0, GL_AMBIENT, light_Ambient );
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_Diffuse );
    glLightfv(GL_LIGHT0, GL_POSITION, light_Position );
}

void ControlMovimientoRaton( GLsizei x, GLsizei y )
{
    angle1= (float)x*0.3;
    angle2--(float)y*0.3;
}

// espefera
void pelota()
{
    glNewList(1, GL_COMPILE);
    quadobj = gluNewQuadric();
    gluQuadricDrawStyle( quadobj, GLU_FILL);
    gluQuadricNormals( quadobj, GLU_SMOOTH);
    gluSphere(quadobj, 0.4, 5, 5);
    gluDeleteQuadric(quadobj);
    glEndList();
}
```

```

animación
void idle(void)
{
    glutPostRedisplay();
    vangular=vangular-2*pi*frecuencia;
    desf=desf+E;
    if(desf>=15)
    {
        desf=-4; Afoton=0; dese=0;
    }
    if(desf>2.5) Afoton=foton;
}

void LeerArchivo()
{
    char LOf[100];
    char fotonf[100];
    char electronf[100];
    ifstream texto("Pfinal.txt");

    texto >> LOf;
    texto >> fotonf;
    texto >> electronf;
    texto.close
    LO=atof(LOf)*1e-10;
    foton=atof(fotonf);
    electron=atof(electronf);
    w=3.1416*2*(1/LO);
    E=h*w;
}

void dibujo()
{
    glNewList(2, GL_COMPILE);
    for (int i=0; i<50; i++)
    {
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, BlueMaterial
);
        glMatrixMode(GL_MODELVIEW);
        glPushMatrix();
        glRotatef(pos, 0.0, 0.0, 1.0);
        glTranslatef(3, 0, 0);
        glCallList(1);
        glPopMatrix();
        pos=pos+20;
    }
    for (int i=0; i< 20; i++)
    {
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, BlueMaterial
);
        glMatrixMode(GL_MODELVIEW);
        glPushMatrix();
        glRotatef(pos, 0.0, 0.0, 1.0);
        glTranslatef(2, 0, 0);
        glCallList(1);
        glPopMatrix();
        pos=pos+20;
    }
}

```

```

        glEndList();
        glNewList(3, GL_COMPILE);
        pos=-6;

for (int i=0; i< 6; i++)
{
    glRotatef(Afoton,0.0,0.0,1.0);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, RedMaterial );
    glMatrixMode(GL_MODELVIEW);
    glPushMatrix();
    glTranslatef(pos,0,0);
    glCallList(1);
    glPopMatrix();
    pos=pos-1;
}
glEndList();
}
void EjesReferencia()
{
    glBegin (GL_LINES);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, RedMaterial );
    glVertex3f ( -20.0, 0.0, 0.0);
    glVertex3f ( 20.0, 0.0, 0.0);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, RedMaterial );
    glVertex3f ( 0.0, 0.0, 0.0);
    glVertex3f ( 0.0,20.0, 0.0);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, RedMaterial );
    glVertex3f ( 0.0, 0.0, 0.0);
    glVertex3f ( 0.0, 0.0,20.0);
    glEnd();
}
void RenderScene(void)
{
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt( angle1, 0, -angle2, 1, 1, -5,0, 1, 0);
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    EjesReferencia();
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, RedMaterial );
    LeerArchivo();

// electrón
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, BlueMaterial );
    glMatrixMode(GL_MODELVIEW);
    glPushMatrix();
    glRotatef(0,0.0,0.0,1.0);
    glTranslatef(0,0,0);
    glCallList(1);
    glPopMatrix();
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, BlueMaterial );
    glMatrixMode(GL_MODELVIEW);
    glPushMatrix();
    glRotatef(vangular,0.0,0.0,1.0);
    glCallList(2);
    // fotones
    glPushMatrix();
    glRotatef(-45,0.0,0.0,1.0);

```

```

glTranslatef(desf,0,0);
glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, BlueMaterial );
glMatrixMode(GL_MODELVIEW);
glPushMatrix();
glTranslatef(0.5,-1,0);
pos=-6;
for (int i=0; i<100; i++)
{
    glRotatef(Afoton,0.0,0.0,1.0);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, RedMaterial );
    glMatrixMode(GL_MODELVIEW);
    glPushMatrix();
    glTranslatef(pos,0,0);
    glCallList(1);
    glPopMatrix();
    pos=pos-1;
}
glPopMatrix();
glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, BlueMaterial );
glMatrixMode(GL_MODELVIEW);
glPushMatrix();

glTranslatef(0.5,1,0);
pos=-6;
for (int i=0; i<100; i++)
{
    glRotatef(Afoton,0.0,0.0,1.0);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, RedMaterial );
    glMatrixMode(GL_MODELVIEW);
    glPushMatrix();
    glTranslatef(pos,0,0);
    glCallList(1);
    glPopMatrix();
    pos=pos-1;
}
glPopMatrix();
glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, BlueMaterial );
glMatrixMode(GL_MODELVIEW);
glPushMatrix();
glTranslatef(0.5,0,0);
pos=-6;
for (int i=0; i<100; i++)
{
    glRotatef(Afoton,0.0,0.0,1.0);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, RedMaterial );
    glMatrixMode(GL_MODELVIEW);
    glPushMatrix();
    glTranslatef(pos,0,0);
    glCallList(1);
    glPopMatrix();
    pos=pos-1;
}
glPopMatrix();
glPopMatrix();
glFlush();
glutSwapBuffers();
}

```

```
int main(void)
{
    int IdeWindow;
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH); //NOTE
    glutInitWindowSize(800,600);
    glutInitWindowPosition(300,100);
    IdeWindow=glutCreateWindow("Compton");
    glutDisplayFunc(RenderScene);
    glutIdleFunc( idle );
    glutMotionFunc ( ControlMovimientoRaton );
    glutSetCursor(GLUT_CURSOR_CROSSHAIR);
    glClearColor(0.5f, 0.0f, 0.7f, 0.0f);
    glEnable(GL_DEPTH_TEST);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(15,800/600,.1,1000);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(50,0,50, 0,0,0, 0,1,0);
    luces();
    pelota();
    EjesReferencia();
    LeerArchivo();
    dibujo();
    glutMainLoop();
    glutDestroyWindow(IdeWindow);
    return 0;
}
```