

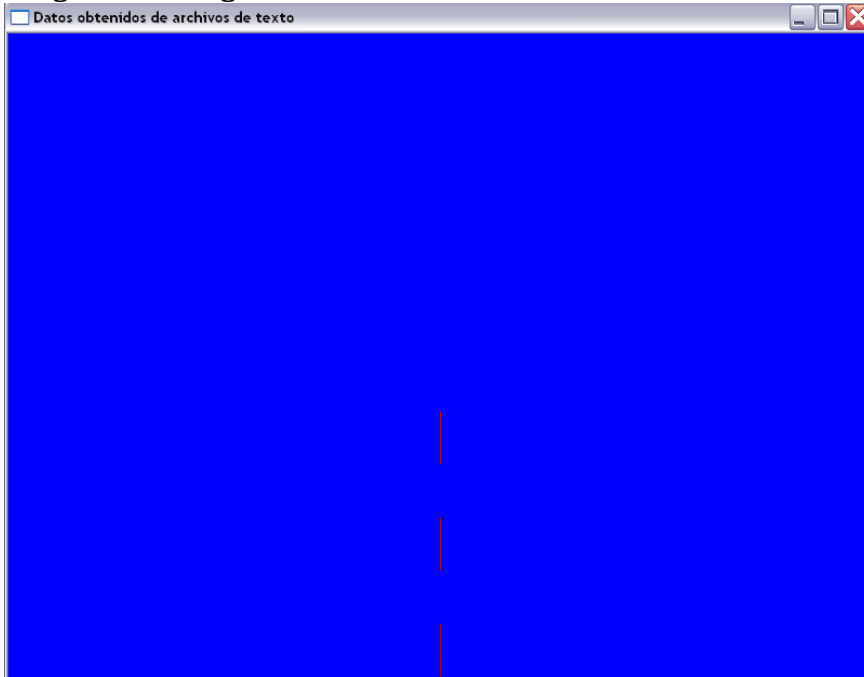
**Práctica :** Archivos de texto con Opendgl

---

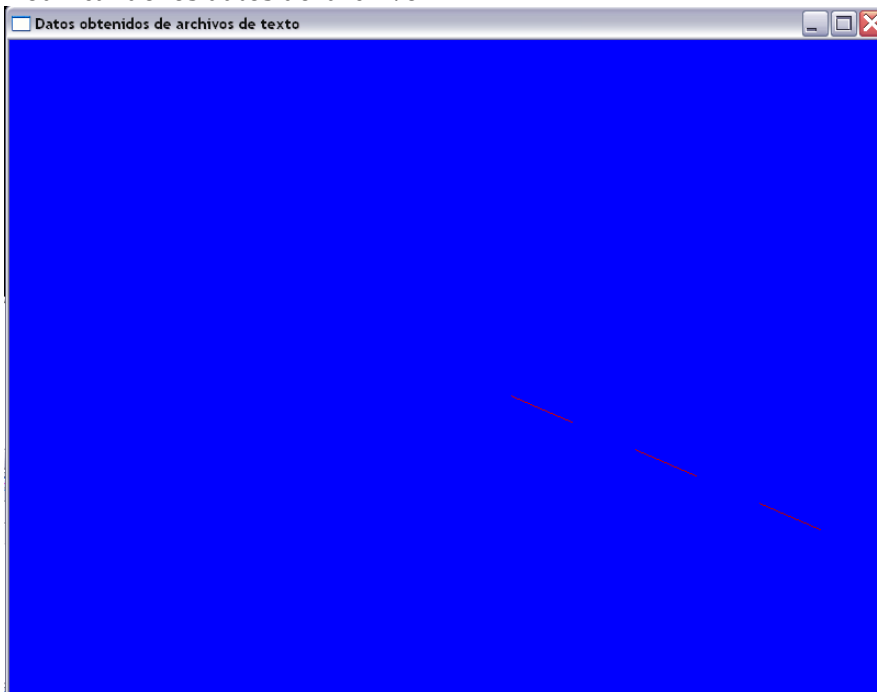
**Objetivo:**

Leer un archivo txt que contenga datos numéricos desde Opendgl y graficarlos en la ventana de muestra.

**Imágenes de las graficas obtenidas:**



**Modificando los datos del archivo**



## Código:

```
#include <gl/openglut.h>
#include <fstream>
//declaracion de variables

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 };

int punto[];
void inicializacion()
{
    glutInitWindowSize(800,600);
    glutInitWindowPosition(100,100);
    glutCreateWindow("Datos obtenidos de archivos de texto");
}

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 materiales()
{
    glMaterialfv (GL_FRONT, GL_AMBIENT_AND_DIFFUSE, material );
    glPolygonMode (GL_FRONT_AND_BACK, GL_FILL);
    glShadeModel (GL_FLAT);
}

void EjesReferencia()
{
    int punto[10];
    int cont=0 ;

    //Función que abre el archivo hola.txt
    std::ifstream texto("num.txt");
    //Leemos el contenido del archivo y lo guardamos en la
variable punto
    while(!texto.eof()){
        texto >> punto[cont];
        cont++;
    }
    //Cerramos el archivo
    texto.close();

    glBegin (GL_LINES);
    //Con los datos obtenidos creamos lineas
    for(int j=0;j<cont;j++){
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, RedMaterial
);
    }
};
```

```

        glVertex3f ( 0, 0, punto[j]);
    }

    glEnd();

}

void RenderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
    luces();
    EjesReferencia();
    materiales();
    glFlush();

}

void camara()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-5,5,-5,5,-10,10);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0,0,0, 15,15,15, 0,1,0);

}

int main(void)
{
    inicializacion();
    glutDisplayFunc(RenderScene);
    glClearColor(0.0f, 0.0f, 1.0f, 1.0f);

    glEnable(GL_DEPTH_TEST);
    camara();
    glutMainLoop();

    return 0;
}

```

### Conclusiones:

Por medio de esta práctica podemos observar que podemos obtener archivos que contengan toda una estructura, como dibujos, objetos, ayudándonos de un editor, en este caso se hizo de manera manual, pero nos ahorraría mucho tiempo obtener archivos con los vértices de un dibujo y cargarlo directamente como lo hicimos en el ejemplo.