

## PRACTICA N.-7

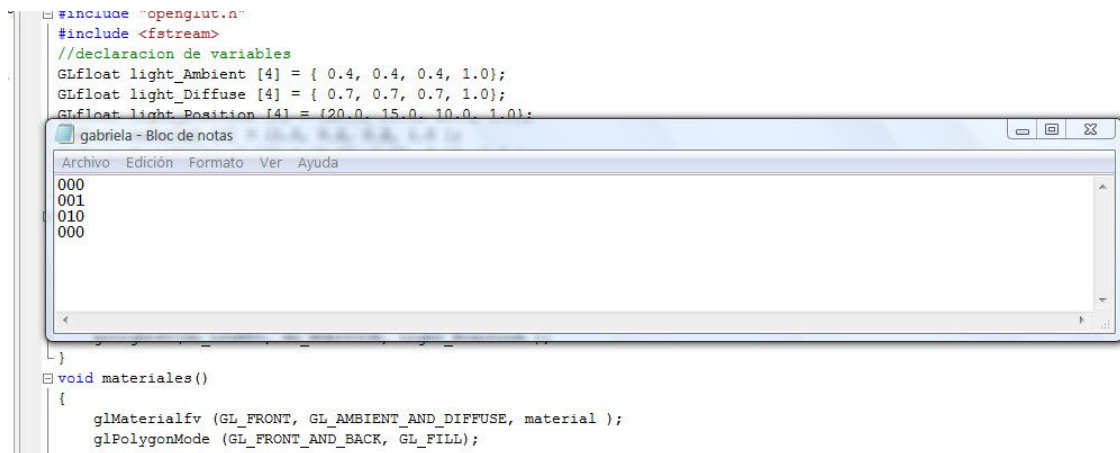
### “IMPORTACIÓN DE UN ARCHIVO CON EXTENSIÓN .TXT A OPENGL”

#### OBJETIVO:

Que el alumno aprenda como importar un archivo de texto al entorno de opengl para poder modelar figuras utilizando otro método de obtención de coordenadas.

#### DESARROLLO:

1. Lo primero que se tiene que hacer es crear el archivo con extensión .txt en el cual pondremos algunas coordenadas para el creado de figuras.



```
#include "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};

}

void materiales()
{
    glMaterialfv (GL_FRONT, GL_AMBIENT_AND_DIFFUSE, material );
    glPolygonMode (GL_FRONT_AND_BACK, GL_FILL);
}
```

gabriela - Bloc de notas

```
Archivo Edición Formato Ver Ayuda
000
001
010
000
```

2. En este caso el archivo tiene el nombre de “gabriela” y será mandado llamar mediante la primitiva : `std::ifstream texto("gabriela.txt")`. Con esto el archivo ya esta siendo leído por opengl, ahora falta asignarle a alguna variable los valores guardados en el.
3. Para leer el contenido del archivo usamos: `while(!texto.eof())`.

```
void Triangulo()
{
    int punto[10];
    int conta=0 ;

    std::ifstream texto("gabriela.txt");
    //Leemos el contenido del archivo y lo guardamos en la variable punto
    while(!texto.eof()){
        texto >> punto[conta];
        conta++;
    }
    //Cerramos el archivo
    texto.close();
    glLineWidth(6.0);
    glEnable(GL_LINE_STIPPLE);
    glColor3f(0.0, 0.0, 0.0);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINES);
    glBegin(GL_POLYGON);
}
```

4. El contenido de cada uno de los valores del archivo .txt sera guardado en una varibale con el nombre de "punto".

```

ution 'texto' (1 pro
texto
  Header Files
  Resource Files
  Source Files
  texto.cpp
}

//Con los datos obtenidos creamos los vertices
for(int k=0;k<conta;k++){

    glVertex3f ( 0, 0, punto[k]);
    glVertex3f ( 0, punto[k], 0);
    glVertex3f (0,punto[k],0);
    glVertex3f ( punto[k], punto[k], punto[k]);
    glVertex3f ( 0, 0, punto[k]);
    glVertex3f ( 0, punto[k], 0);
    glVertex3f ( 0, 0, punto[k]);

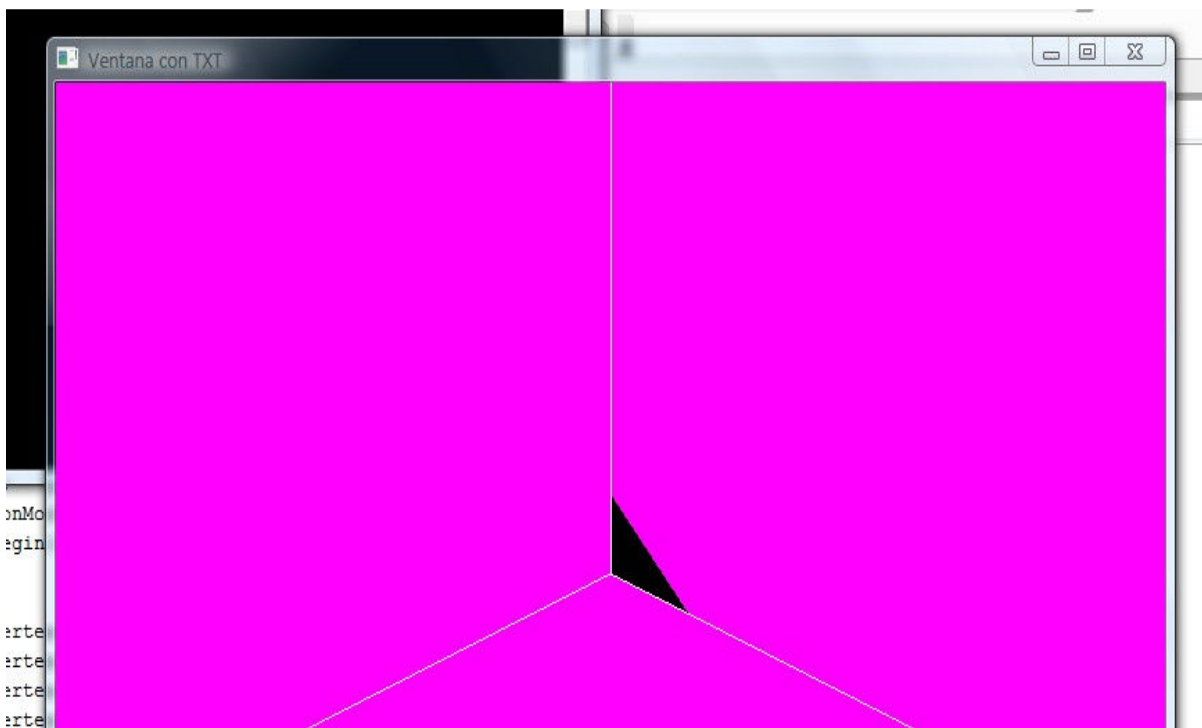
    glEnd();
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glBegin(GL_POLYGON);

    glVertex3f ( 0, 0, punto[k]);
    glVertex3f ( 0, punto[k], 0);
    glVertex3f (0,punto[k],0);
    glVertex3f ( punto[k], punto[k], punto[k]);
    glVertex3f ( 0, 0, punto[k]);
    glVertex3f ( 0, punto[k], 0);
    glVertex3f ( 0, 0, punto[k]);

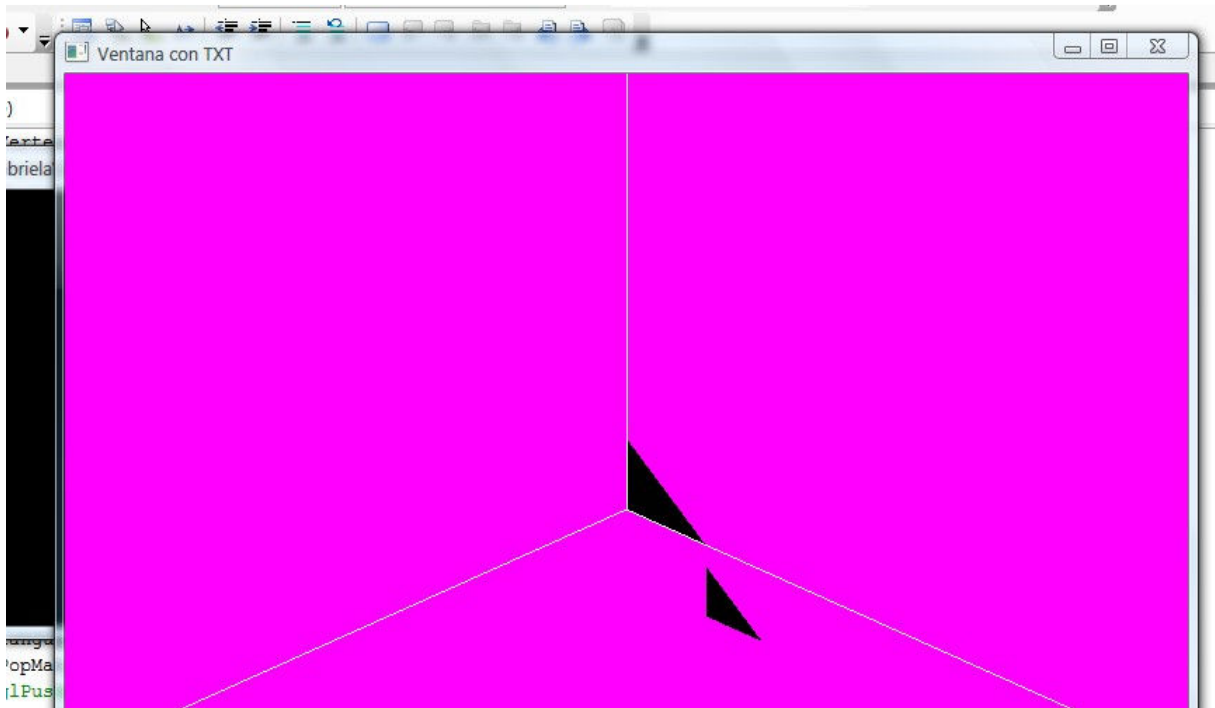
    glEnd();
}
}

```

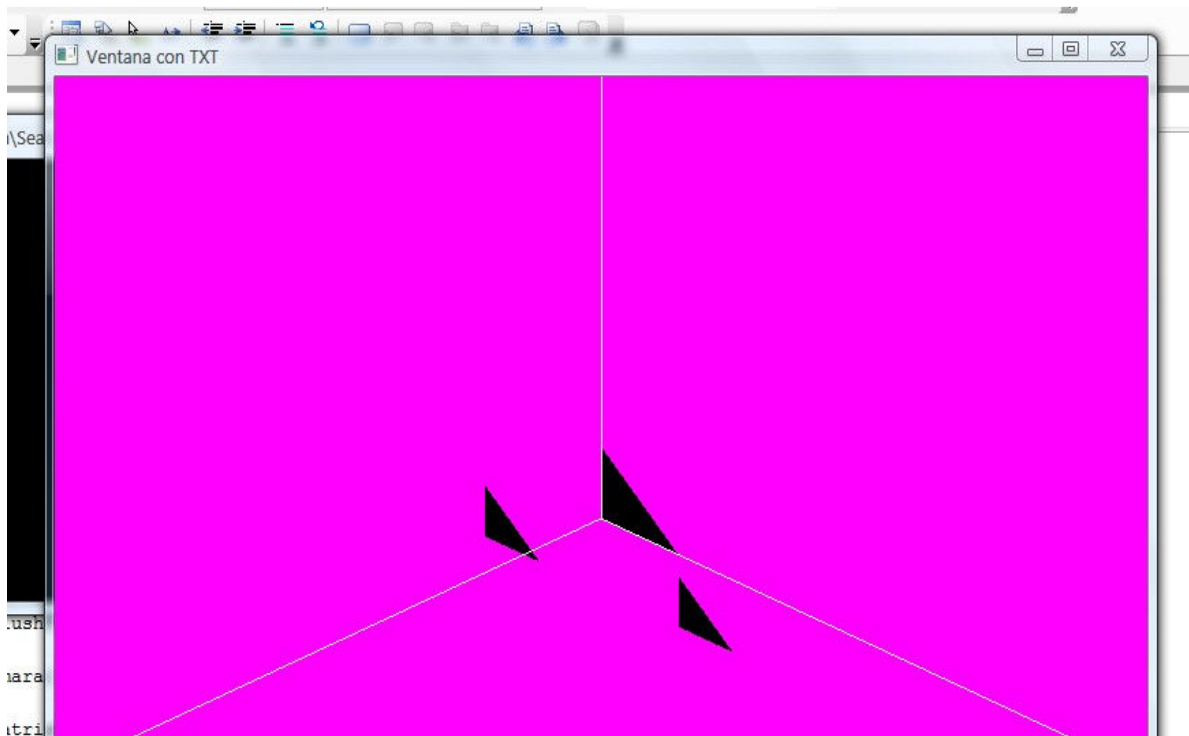
5. Luego para cerrar el archivo se utiliza la sentencia: `texto.close()`, ya que no es necesario que opengl continúe con éste abierto, ya que los valores ya fueron guardados.
6. Y por último se ven los resultados y se comienza a jugar las coordenadas.



Se dibujo un triangulo en el centro de coordenadas



*Ahora 2 triángulos, uno trasladado en "z"*



*Tres triángulos en escena*

## CONCLUSIÓN:

En esta práctica pude aprender a utilizar un archivo txt para poder realizar figuras, así ya contamos con una nueva herramienta para el modelado dentro de opengl, además me pareció muy interesante observar que la interacción entre ellos se puede decir que es relativamente fácil, lo único que complica esta práctica fue la obtención de coordenadas en el archivo txt, y después la forma de utilizar dentro de opengl.