

Ejercicio 3

Graficación por Computadora

April 25, 2019

Para este ejercicio el alumno generará un programa que cargue y muestre un modelo proveniente de un archivo Wavefront .obj, para leer de un archivo en C se utilizarán las funciones que se encuentran en la biblioteca estándar de C para entrada y salida `<stdio.h>`, si se usa C++ se puede usar también `<iostream>` que es su equivalente.

1 Wavefront .obj

Los archivos Wavefront .obj son archivos que definen la geometría tridimensional de muchos o varios objetos.

Mostramos un formato simplificado:

```
# Comentario
# lista de vertices geometricos (x, y, z)
v 0.123 0.313 -2.012
v ...
...
# lista de las coordenadas (x,y) de las texturas
# para cada vertice geometrico
vt 0.2 1
vt ...
...
# lista de las normales (x, y, z) para cada
# vertice geometrico
vn 0.0 0.0 1.0
vn ...
...
# lista de caras (pueden no ser triangulares)
f 1 2 3
f ...
...
```

Noten que las líneas que contienen información geométrica tienen un formato estándar, por lo que se puede usar la función `fscanf()` para obtener los valores.

1.1 Vértices

Las líneas que tienen vértices comienzan con una `v` seguida de un espacio y tres valores de punto flotante (puede tener cuatro para el componente w), claramente termina con un salto de línea, es necesario guardar estos vértices. Se recomienda guardar en un arreglo dinámico (usando `malloc()`),

nota que el número de vértices no viene especificado en el archivo por lo que se tendrían que contar antes de crear este arreglo, pero si se desea se puede usar un arreglo global con un número de vértices muy grande (considera que esta opción gasta memoria).

Estos archivos pueden venir con las normales (**vn**) y coordenada de texturas (**vt**) para cada vértice, se le da un punto extra el agregarle el manejo de esta información a la función, una recomendación para esto es crear una estructura **Vertex** donde se pueda guardar un vértice, su normal y su coordenada de textura ya sólo se crea un arreglo de estas estructuras.

1.2 Caras

Las líneas que tienen caras comenzand con una **f** seguida de un espacio y tres valores enteros para triángulos (o cuatro valores para cuadrilateros, etc.), para simplificar la función sólo se manejarán las caras triangulares. Los valores enteros los guardaremos en otro arreglo (aplica lo mismo que para los vértices). Noten que cada entero representa el índice de un vértice, en un .obj el rango de los n vértices es $[1, n]$, como el rango de nuestros arreglos es $[0, n - 1]$ es necesario decrementar por uno el valor de cada índice.

Si se están manejando las normales y coordenadas de texturas entonces las lineas se verán así:

```
f 1/2/3 2/4/3 3/4/2
```

Este formato también cambia si sólo se tienen las normales o si sólo se tienen las coordenadas de texturas, dado que no necesitamos las normales y texturas para dibujar al modelo, estas las ignoraremos, de igual manera debemos obtener los indices de los vértices, los formatos se encuentran en https://en.wikipedia.org/wiki/Wavefront_.obj_file#Face_elements.

1.3 Dibujado

La parte más sencilla, al ya tener nuestro modelo en memoria simplemente vamos a recorrer el arreglo de las caras y obtener los tres vértices a partir de sus índices, de aquí se podrá dibujar en pantalla usando **GL_TRIANGLES**. Si se desea pueden agregar eventos de teclado para rotar el objeto y poder entender más la figura ya que no tendrá sombreado.

2 Entrega

Se entregará un comprimido .zip con el Makefile y los archivos main.c y/o main.h comentados/documentados junto con algún modelo en .obj que usaron de prueba, además agregarán un README.txt donde especifiquen su nombre completo, número de cuenta y, si se desea, algún comentario sobre el ejercicio. Subir al Moodle el comprimido con el siguiente formato:

<Apellido Paterno><Apellido Materno>_03.zip