Computación Gráfica

Práctico 1
Introducción a OpenGL

1 Problemas obligatorios

1. Utilizando la función ***glGetString***, imprima por consola el número de versión de *OpenGL* que utiliza su sistema.
2. Dibuje el triángulo formado por los vértices [(-1,-1,-2), (1,-1,-2), (0,1,-2)].
3. Dibuje el mismo triángulo, pero de forma que cada vertice tenga un color diferente.
4. Dibuje el mismo triángulo, pero reducido a la mitad de su tamaño.

   Nota*:* No modifique las coordenadas de los vértices, utilice transformaciones.
5. Traslade el triángulo al punto (0, 3, -5).
6. Dibuje el mismo triángulo, pero rotado 30° para cada uno de los 3 ejes por separado.
7. Genere un game loop que en cada itereción incremente la rotación del triángulo en 0.1°.
8. Dibuje el modelo definido por el archivo ***knight.obj***[1] disponible en webasignatura. El formato del archivo *OBJ* se encuentra definido en el anexo I de este documento. Cada triángulo del modelo debe ser dibujado usando un tono de gris randómico en sus vértices.

   Nota: Para parsear el archivo *obj*, puede utilizar las funciones *fgets* (definida en *stdio.h*), *strcmp, strtok* (definidas en *string.h*), *atof* y *atoi* (definidas en *stdlib.h*).
9. Dibuje el modelo en formato *Malla de alambre*.
10. Permita rotar el modelo horizontalmente al oprimir las teclas ← y →.

2 Problemas complementarios

1. Dibuje un modelo utilizando *Triangle Strips* y otro utilizando *Triangle Fans*.
2. Dibuje un modelo utilizando cuadriláteros (*GL_QUADS)* en lugar del triángulos.
3. Implemente su propio framework de manejo de Matrices, limitandose a utilizar unicamente las funciones de *OpenGL* ***glLoadIdentity*** y ***glLoadMatrixf***.

   Nota: Tenga en cuenta que *OpenGL* almacena las matrices de 4x4 por columnas, en arrays de 16 elementos.
4. Implemente todos los ejercios utilizando *Vertex Arrays* en lugar del modo directo.

1 http://webasignatura.ucu.edu.uy/pluginfile.php?file=%2F15542%2Fmod_folder%2Fcontent%2F0%2FArchivos%2FModels%2FOBJ%2Fknight.obj&forcedownload=1

3 Anexo I

Los modelos *OBJ* son archivos de texto definen geometrías 3D. Cáda línea del archivo se compone de un identificador y una serie de valores. El identificador determina la propiedad del modelo que se está definiendo, y el resto de la línea el/los valor/es de esta propiedad.

Estos son los identificadores que utilizaremos:

- *#comentario.* Los comentarios comienzan con el caracter # y no configuran ninguna propiedad del modelo, deben ser ignorados durante el pareseo del archivo.
- *o nombre*. La línea que comienza con el carcter 'o' seguido de un 'nombre' determina el nombre del modelo. Esta línea es opcional.
- *v x y z.* Las líneas que comienzan con el caracter 'v' definen un vértice del modelo, y se encuentran seguidas de 3 valores float, separados por espacios, correspondientes a las coordenadas 3D del vértice.
- *f v0 v1 v2 [v4]*. Las líneas que comienzan con el caracter 'f' definen una cara del modelo, y se encuentran seguidas, generalmente, por 3 índices enteros correspondientes a los vértices que definen el triángulo. Opcional, puede encontrarse un 4to índice, en el caso de que la cara sea un cuadrilátero en lugar de un triángulo.
- ***Nota***: Los índices a los vértices comienzan en el número 1, no el 0.

Ejemplo: *box.obj*

```
o box

# 8 Vertices (vertexs)
v -0.500000 -0.500000 -0.500000
v 0.500000 -0.500000 -0.500000
v 0.500000 -0.500000 0.500000
v -0.500000 -0.500000 0.500000
v -0.500000 0.500000 0.500000
v 0.500000 0.500000 0.500000
v 0.500000 0.500000 -0.500000
v -0.500000 0.500000 -0.500000

# 12 Poligonos (faces)
f 3 7 6
f 7 3 2
f 1 5 8
f 5 1 4
f 1 7 2
f 7 1 8
f 5 3 6
f 3 5 4
f 7 5 6
f 5 7 8
f 1 3 4
f 3 1 2

#Fin Archivo
```