

UNIVERSIDAD CENTRAL DEL ECUADOR

Facultad de Ingeniería en Ciencias Físicas y Matemática

Ingeniería en Computación Gráfica

Programación Gráfica II

GLU Y GLUT

Pamela Barrionuevo

Abril 17, 2019

**Resumen**

OpenGL dispone de varias librerías y funciones para que se pueda compilar dicho programa con figuras 2D y 3D, es por eso que esta consulta se incluirá páginas de referencia para todas las funciones de la Biblioteca de Utilidad OpenGL, como son las funciones GLU y GLUT.

AsGLU es la biblioteca de utilidades OpenGL. Este es un conjunto de funciones para crear mapas de texturas desde una imagen base, mapear coordenadas entre la pantalla y el espacio de objetos, y dibujar superficies cuadráticas y NURBS. GLU 1.2 es la versión de GLU que va con OpenGL 1.1. GLU 1.3 está disponible e incluye nuevas capacidades correspondientes a las nuevas funciones de OpenGL 1.2.

Además, GLUT es el OpenGL Utility Toolkit, un conjunto de herramientas independiente del sistema de ventanas para escribir programas OpenGL. Implementa una interfaz de programación de aplicaciones (API) simple para ventanas para OpenGL. GLUT hace que sea mucho más fácil aprender y explorar la Programación OpenGL.

**Tabla de Contenidos**

[Biblioteca de GLU 5](#_Toc6349924)

[Funciones GLU 5](#_Toc6349925)

[gluBeginCurve ,gluEndCurve 5](#_Toc6349926)

[gluBeginPolygon ,gluEndPolygon 5](#_Toc6349927)

[gluBeginSurface ,gluEndSurface 5](#_Toc6349928)

[gluBeginTrim , gluEndTrim 5](#_Toc6349929)

[gluBuild1DMipmaps 5](#_Toc6349930)

[gluBuild2DMipmaps 5](#_Toc6349931)

[cilindro azul 2](#_Toc6349932)

[gluDeleteNurbsRenderer 2](#_Toc6349933)

[gluDeleteQuadric 2](#_Toc6349934)

[GluDeleteTess 2](#_Toc6349935)

[GluDisk 2](#_Toc6349936)

[gluErrorString 2](#_Toc6349937)

[gluGetNurbsProperty 2](#_Toc6349938)

[gluGetString 2](#_Toc6349939)

[gluGetTessProperty 2](#_Toc6349940)

[gluLoadSamplingMatrices 2](#_Toc6349941)

[GluLookAt 2](#_Toc6349942)

[gluNewNurbsRenderer 2](#_Toc6349943)

[gluNewQuadric 2](#_Toc6349944)

[GluNewTess 2](#_Toc6349945)

[GluNextContour 3](#_Toc6349946)

[*gluNurbsCallback* 3](#_Toc6349947)

[GluNurbsCurve 3](#_Toc6349948)

[propiedad de GluNurbs 3](#_Toc6349949)

[GluNurbsSuperficie 3](#_Toc6349950)

[GluOrtho2D 3](#_Toc6349951)

[GluPartialDisk 3](#_Toc6349952)

[gluPerspectiva 3](#_Toc6349953)

[gluPickMatrix 3](#_Toc6349954)

[gluProject 3](#_Toc6349955)

[gluPwlCurve 3](#_Toc6349956)

[*gluQuadricCallback* 3](#_Toc6349957)

[gluQuadricDrawStyle 3](#_Toc6349958)

[gluQuadricNormals 4](#_Toc6349959)

[Orientación gluQuadric 4](#_Toc6349960)

[gluQuadricTexture 4](#_Toc6349961)

[gluScaleImage 4](#_Toc6349962)

[gluSphere 4](#_Toc6349963)

[gluTessBeginContour ,gluTessEndContour 4](#_Toc6349964)

[gluTessBeginPolygon ,gluTessEndPolygon 4](#_Toc6349965)

[*GluTessCallback* 4](#_Toc6349966)

[GluTessNormal 4](#_Toc6349967)

[GluTessProperty 4](#_Toc6349968)

[gluTessVertex 4](#_Toc6349969)

[gluUnProject 4](#_Toc6349970)

[BIBLIOTECA GLUT 5](#_Toc6349971)

[Qué es GLUT? 5](#_Toc6349972)

[GLUT según su creador 5](#_Toc6349973)

[Primitivas de objetos predefinidos 6](#_Toc6349974)

[- glutWireSphere(radius, slices, stacks), glutSolidSphere(radius, slices, stacks) 6](#_Toc6349975)

[- glutWireCube(size), glutSolidCube(size) 6](#_Toc6349976)

[- glutWireCone(base, height, slices, stacks), glutSolidCone(base, height, slices, stacks) 6](#_Toc6349977)

[- glutWireDodecahedron(void), glutSolidDodecahedron(void) 6](#_Toc6349978)

[- glutWireOctahedron(void), glutSolidOctahedron(void) 6](#_Toc6349979)

[- glutWireTetrahedron(void), glutSolidTetrahedron(void) 7](#_Toc6349980)

[- glutWireIcosahedron(void), glutSolidIcosahedron(void) 7](#_Toc6349981)

[- glutWireTeapot(void), glutSolidTeapot(void) 7](#_Toc6349982)

[Estructura de un programa usando la GLUT 7](#_Toc6349983)

[void glutDisplayFunc( void (\*func) (void)) 8](#_Toc6349984)

[void glutReshapeFunc(void (\*func) (int width, int height)) 8](#_Toc6349985)

[Control del mouse 8](#_Toc6349986)

[void glutMouseFunc( void (\*func) (int button, int state, int x, int y)) 8](#_Toc6349987)

[void glutMotionFunc(void (\*func) (int x, int y)) 8](#_Toc6349988)

[Definición de menús 8](#_Toc6349989)

[int glutCreateMenu(void (\*func) (int value) 8](#_Toc6349990)

[void glutSetMenu(int menu) 8](#_Toc6349991)

[void glutAddMenuEntry(char \*name, int value) 8](#_Toc6349992)

[void glutAddSubMenu(char \*name, int menu) 8](#_Toc6349993)

[void glutChangeToMenuEntry(int entry, char \*name, int value) 8](#_Toc6349994)

[void glutAttachMenu(int button) 9](#_Toc6349995)

[CONCLUSIONES 9](#_Toc6349996)

[RECOMENDACIONES 9](#_Toc6349997)

[Lista de referencias 10](#_Toc6349998)

## Biblioteca de GLU

La biblioteca GL Utilities (GLU) es un conjunto de rutinas diseñadas para complementar el sistema de gráficos OpenGL al proporcionar soporte para mipmapping. Manipulación de matrices, teselación de polígonos, cuadráticas, NURBS y manejo de errores. Las rutinas de Mipmapping incluyen escalado de imagen y mipmap automático. Generando una variedad de funciones de manipulación de matrices construyen proyección y visualización de matrices, o proyecto de vértices de un sistema de coordenadas a otro.

Las rutinas de teselación de polígonos convierten los polígonos cóncavos en triángulos para facilitar representación. El soporte de Quadrics hace unos cuantos cuadráticos básicos como las esferas y conos. El código NURBS asigna curvas NURBS complicadas y recortadas superficies en evaluadores más simples de OpenGL. Por último, una rutina de búsqueda de errores. Traduce los códigos de error de OpenGL y GLU en cadenas. Las rutinas de biblioteca GLU pueden llamar rutinas de biblioteca OpenGL. Por lo tanto, un contexto OpenGL debería actualizarse antes de llamar a cualquier función de GLU. De lo contrario un error puede ocurrir en OpenGL. (Khronos, 2019)

## Funciones GLU

A continuación encontraremos las funciones de GLU, las cuales tendrán acceso al dar clic sobre ellas y que se abra más información de cada función en internet.

| **Función** | **Descripción** |
| --- | --- |
| [**gluBeginCurve**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glubegincurve) ,**[gluEndCurve](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluendcurve)** | Delimite una definición de curva de B-Spline racional no uniforme ( [NURBS](https://docs.microsoft.com/en-us/windows/desktop/opengl/using-nurbs-curves-and-surfaces)). |
| [**gluBeginPolygon**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glubeginpolygon) ,**[gluEndPolygon](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluendpolygon)** | Delimitar una descripción de polígono. |
| [**gluBeginSurface**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glubeginsurface) ,**[gluEndSurface](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluendsurface)** | Delimitar una definición de superficie NURBS. |
| [**gluBeginTrim**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glubegintrim) ,[**gluEndTrim**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluendtrim) | Delimitar una definición de bucle de recorte NURBS. |
| [**gluBuild1DMipmaps**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glubuild1dmipmaps) | Crea mipmaps 1-D. |
| [**gluBuild2DMipmaps**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glubuild2dmipmaps) | Crea mipmaps 2-D. |
| [**cilindro azul**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glucylinder) | Dibuja un cilindro. |
| [**gluDeleteNurbsRenderer**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gludeletenurbsrenderer) | Destruye un objeto NURBS. |
| [**gluDeleteQuadric**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gludeletequadric) | Destruye un objeto cuadrático. |
| [**GluDeleteTess**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gludeletetess) | Destruye un objeto de teselación. |
| [**GluDisk**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gludisk) | Dibuja un disco. |
| [**gluErrorString**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluerrorstring) | Produce una cadena de error a partir de un código de error OpenGL o GLU.La cadena de error es solo ANSI. |
| [**gluGetNurbsProperty**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glugetnurbsproperty) | Recupera una propiedad NURBS. |
| [**gluGetString**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glugetstring) | Recupera una cadena que describe el número de versión de GLU o las llamadas de extensión de GLU compatibles. |
| [**gluGetTessProperty**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glugettessproperty) | Recupera una propiedad de objeto de teselación. |
| [**gluLoadSamplingMatrices**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluloadsamplingmatrices) | Carga matrices de muestreo y de selección de NURBS. |
| [**GluLookAt**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glulookat) | Define una transformación de visualización. |
| [**gluNewNurbsRenderer**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glunewnurbsrenderer) | Crea un objeto NURBS. |
| [**gluNewQuadric**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glunewquadric) | Crea un objeto cuadrático. |
| [**GluNewTess**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glunewtess) | Crea un objeto de teselación. |
| [**GluNextContour**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glunextcontour) | Marca el comienzo de otro contorno. |
| [gluNurbsCallback](https://docs.microsoft.com/en-us/windows/desktop/opengl/glunurbs) | Define una devolución de llamada para un objeto NURBS. |
| [**GluNurbsCurve**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glunurbscurve) | Define la forma de una curva NURBS. |
| [**propiedad de GluNurbs**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glunurbsproperty) | Establece una propiedad NURBS. |
| [**GluNurbsSuperficie**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glunurbssurface) | Define la forma de una superficie NURBS. |
| [**GluOrtho2D**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluortho2d) | Define una matriz de proyección ortográfica 2-D. |
| [**GluPartialDisk**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glupartialdisk) | Dibuja un arco de un disco. |
| [**gluPerspectiva**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluperspective) | Establece una matriz de proyección en perspectiva. |
| [**gluPickMatrix**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glupickmatrix) | Define una región de picking. |
| [**gluProject**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluproject) | Asigna las coordenadas del objeto a las coordenadas de la ventana. |
| [**gluPwlCurve**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glupwlcurve) | Describe una curva de recorte NURBS lineal por partes. |
| [gluQuadricCallback](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluquadric) | Define una devolución de llamada para un objeto cuadrático. |
| [**gluQuadricDrawStyle**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluquadricdrawstyle) | Especifica el estilo de dibujo deseado para las cuadráticas. |
| [**gluQuadricNormals**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluquadricnormals) | Especifica qué tipo de normales se utilizarán para las cuadráticas. |
| [**Orientación gluQuadric**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluquadricorientation) | Especifica orientación interior o exterior para cuadráticas. |
| [**gluQuadricTexture**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluquadrictexture) | Especifica si las cuadráticas deben ser texturizadas. |
| [**gluScaleImage**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluscaleimage) | Escala una imagen a un tamaño arbitrario. |
| [**gluSphere**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glusphere) | Dibuja una esfera. |
| [**gluTessBeginContour**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glutessbegincontour) ,**[gluTessEndContour](https://docs.microsoft.com/en-us/windows/desktop/opengl/glutessendcontour)** | Delimitar una descripción del contorno. |
| [**gluTessBeginPolygon**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glutessbeginpolygon) ,**[gluTessEndPolygon](https://docs.microsoft.com/en-us/windows/desktop/opengl/glutessendpolygon)** | Delimitar una descripción de polígono. |
| [GluTessCallback](https://docs.microsoft.com/en-us/windows/desktop/opengl/glutess) | Define una devolución de llamada para un objeto de teselación. |
| [**GluTessNormal**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glutessnormal) | Especifica una normal para un polígono. |
| [**GluTessProperty**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glutessproperty) | Establece la propiedad de un objeto de teselación. |
| [**gluTessVertex**](https://docs.microsoft.com/en-us/windows/desktop/opengl/glutessvertex) | Especifica un vértice en un polígono. |
| [**gluUnProject**](https://docs.microsoft.com/en-us/windows/desktop/opengl/gluunproject) | Asigna las coordenadas de la ventana a las coordenadas del objeto. |

Tabla de Funciones GLU 1

(Satran & Kennedy, 2019)

# BIBLIOTECA GLUT

## Qué es GLUT?

El OpenGL Utility Toolkit ( GLUT ) es una biblioteca de utilidades para programas OpenGL, que principalmente realiza E / S a nivel de sistema con el sistema operativo host . Las funciones realizadas incluyen definición de ventana, control de ventana y monitoreo de entrada de teclado y mouse. También se proporcionan rutinas para dibujar una serie de primitivas geométricas (tanto en modo sólido como de estructura alámbrica), incluidos cubos, esferas y la tetera de Utah. GLUT incluso tiene un soporte limitado para crear ventanas emergentes.

GLUT fue escrito por Mark J. Kilgard , autor de OpenGL Programming para X Window y The CG Tutorial: The Definitive Guide to Programmable Real-Time Graphics mientras trabajaba para Silicon Graphics Inc. (Fandom, 2019)

## GLUT según su creador

Los dos objetivos de GLUT son permitir la creación de un código bastante portátil entre sistemas operativos (GLUT es multiplataforma ) y facilitar el aprendizaje de OpenGL. Comenzar con la programación de OpenGL mientras usa GLUT a menudo toma solo unas pocas líneas de código y no requiere conocimiento de las API de ventanas específicas del sistema operativo . OpenGL UtilityToolkit (GLUT) es una interfaz de programación con enlaces ANSI C y FORTRAN para escribir programas OpenGL independientes del sistema de ventanas. El kit de herramientas admite la siguiente funcionalidad:

* Varias ventanas para la representación de OpenGL.
* Procesamiento de eventos de devolución de llamada.
* Dispositivos de entrada sofisticados.
* Una rutina "inactiva" y temporizadores.
* Un sencillo menú emergente en cascada.
* Rutinas de utilidad para generar varios objetos de marco sólido y alambre.
* Soporte para bitmap y fuentes de trazo.
* Funciones diversas de gestión de ventanas, incluida la gestión de superposiciones.

El autor ha implementado una implementación ANSI C de GLUT para el sistema X Window [13].

Las versiones de Windows NT y OS / 2 de GLUT también están disponibles. (Kilgard, 1996)

## Primitivas de objetos predefinidos

Hay algunos objetos que vamos a renderizar muy a menudo, y que por tanto, ya vienen definidos. Así,

disponemos de las siguientes funciones:

### - glutWireSphere(radius, slices, stacks), glutSolidSphere(radius, slices, stacks)



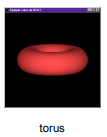
Función GLUT Sphere 1

### - glutWireCube(size), glutSolidCube(size)



Función GLUT cube 1

### - glutWireCone(base, height, slices, stacks), glutSolidCone(base, height, slices, stacks)



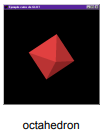
Función GLUT cone y torus 1

### - glutWireDodecahedron(void), glutSolidDodecahedron(void)



Función GLUT dodecahedron 1

### - glutWireOctahedron(void), glutSolidOctahedron(void)



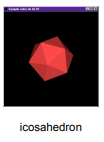
Función GLUT octahedron 1

### - glutWireTetrahedron(void), glutSolidTetrahedron(void)



Función GLUT tetrahedron 1

### - glutWireIcosahedron(void), glutSolidIcosahedron(void)



Función GLUT icosahedron 1

### - glutWireTeapot(void), glutSolidTeapot(void)



Función GLUT teapot 1

(Cañero Morales, 2018)

## Estructura de un programa usando la GLUT

Las aplicaciones que desarrollaremos usando la GLUT se basan en una interficie de ventanas. Este tipo de interficies son muy cómodas para el usuario, pero la programación se complica bastante porque no podemos controlar lo que va a hacer el usuario.

Una manera sencilla de programar este tipo de aplicaciones es utilizando un esquema orientado a

eventos.

Por defecto, la GLUT ya tiene definidas callbacks para cada evento posible, pero, evidentemente,

nosotros querremos personalizar este comportamiento. Para hacer esto, algunas de las funciones de las que disponemos son:

### void glutDisplayFunc( void (\*func) (void))

La función func() se llamará cada vez que haya que redibujar la ventana.

### void glutReshapeFunc(void (\*func) (int width, int height))

La función func(width, height) se llamará cada vez que la ventana cambie de tamaño, y recibirá como parámetros la nueva anchura y altura.

## Control del mouse

De entre los callbacks que nos da la GLUT, son especialmente interesantes para el control del mouse los siguientes:

### void glutMouseFunc( void (\*func) (int button, int state, int x, int y))

La función func(button, state, x, y) se llamará tanto cuando se apriete (state = GLUT\_DOWN)

como cuando se suelte (state = GLUT\_UP) el botón definido por el parámetro button en la posición (x, y).

### void glutMotionFunc(void (\*func) (int x, int y))

La función func(x, y) se llamará cuando el mouse se mueva mientras está pulsado uno de sus

botones.

## Definición de menús

La GLUT provee una manera sencilla de definir menús para nuestra aplicación. Veamos algunas de las herramientas de que disponemos para trabajar con menús:

### int glutCreateMenu(void (\*func) (int value)

Esta función crea un menú (todavía sin opciones), y le asigna la función func(value). Esta función se llamará cada vez que una de las opciones del menú sea seleccionada por el usuario, y recibirá en value el código identificativo de la opción seleccionada. De esta manera, podremos definir que hace cada opción de menú. Devuelve un identificador de menú, que nos servirá cuando tengamos que referirnos a él.

### void glutSetMenu(int menu)

Esta función hace que el menú identificado como menu sea el menú actual. Por defecto, el menú actual es el último que se ha creado.

### void glutAddMenuEntry(char \*name, int value)

Añade una opción de menú al menú actual. Esta opción se llamará name y se identificará por el número value.

### void glutAddSubMenu(char \*name, int menu)

Añade una opción de menú que abrirá un submenú (en lugar de ejecutar directamente un comando, como en el caso anterior). La opción se llamará name y el menú que aparecerá será el identificado como menu (ver función glutCreateMenu).

### void glutChangeToMenuEntry(int entry, char \*name, int value)

Esta función sirve para modificar una opción del menú actual. El parámetro entry nos indica la opción a modificar (por ejemplo, para modificar la primera, entry = 1), name será el nuevo nombre de la opción y value el nuevo identificador que se le pasará a la función controladora de menú.

### void glutAttachMenu(int button)

Esta función hace que el menú actual aparezca cada vez que se pulse el botón del mouse indicado por button (GLUT\_LEFT\_BUTTON, GLUT\_MIDDLE\_BUTTON, GLUT\_RIGHT\_BUTTON). (Cañero Morales, 2018)

# CONCLUSIONES

* Pudimos observar cuan importantes son estas librerías y funciones GLU y GLUT dentro de OpenGL que es una buena elección para generación de efectos visuales, escenas, etc., debido a su portabilidad, estandarización y potencia gráfica.

# RECOMENDACIONES

* Se recomienda tener muy en cuenta al momento de utilizar alguna de las funciones de GLU o GLUT importar las mismas librerías para que así no salga ningún error al momento de poner en práctica alguna función o alguna primitiva que brindan estas grandiosas librerías de OpenGL.

# Lista de referencias

Cañero Morales, C. (2018). *ELAI*. Obtenido de http://www.elai.upm.es/webantigua/spain/Asignaturas/InfoInd/teoria/Apuntes%20de%20OpenGL.pdf

Fandom. (2019). *FANDOM*. Obtenido de https://graphics.fandom.com/wiki/OpenGL\_Utility\_Toolkit

Khronos. (2019). *KHRONOS*. Obtenido de https://www.khronos.org/registry/OpenGL/specs/gl/glu1.3.pdf

Kilgard, M. (23 de February de 1996). *Silicon Graphics, Inc*. Obtenido de https://www.lri.fr/~mbl/ENS/IG2/docs/glut3.6.pdf

Satran, M., & Kennedy, J. (2019). *Microsoft*. Obtenido de https://docs.microsoft.com/en-us/windows/desktop/opengl/glu-functions