

Basic GLUT

Introduction to GLUT function

- ▶ Document
- ▶ Initialization and window
- ▶ Callback Registration
- ▶ Geometric Object Rendering
- ▶ Beginning Event Processing

Document

► <https://www.opengl.org/resources/libraries/glut/spec3/spec3.html>

The OpenGL Utility Toolkit (GLUT) Programming Interface API Version 3

Mark J. Kilgard
Silicon Graphics, Inc.

OpenGL is a trademark of Silicon Graphics, Inc. X Window System is a trademark of X Consortium, Inc. Spaceball is a registered trademark of Spatial Systems Inc.

The author has taken care in preparation of this documentation but makes no expressed or implied warranty of any kind and assumes no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising from the use of information or programs contained herein.

Copyright © 1994, 1995, 1996. Mark J. Kilgard. All rights reserved.

All rights reserved. No part of this documentation may be reproduced, in any form or by any means, without permission in writing from the author.

-
- [Contents](#)
 - [1 Introduction](#)
 - [1.1 Background](#)
 - [1.2 Design Philosophy](#)
 - [1.3 API Version 2](#)
 - [1.4 API Version 3](#)
 - [1.5 Conventions](#)
 - [1.6 Terminology](#)
 - [2 Initialization](#)
 - [2.1 glutInit](#)
 - [2.2 glutInitWindowPosition, glutInitWindowSize](#)
 - [2.3 glutInitDisplayMode](#)
 - [3 Beginning Event Processing](#)
 - [3.1 glutMainLoop](#)
 - [4 Window Management](#)
 - [4.1 glutCreateWindow](#)
 - [4.2 glutCreateSubWindow](#)
 - [4.3 glutSetWindow, glutGetWindow](#)
 - [4.4 glutDestroyWindow](#)
 - [4.5 glutPostRedisplay](#)
 - [4.6 glutSwapBuffers](#)
 - [4.7 glutPositionWindow](#)
 - [4.8 glutReshapeWindow](#)
 - [4.9 glutFullScreen](#)
 - [4.10 glutPopWindow, glutPushWindow](#)
 - [4.11 glutShowWindow, glutHideWindow, glutIconifyWindow](#)
-

Initialization and window

- ▶ void `glutInit`(int *argc, char **argv);
 - ▶ Initializing the GLUT library
 - ▶ Should be called before any GLUT functions
- ▶ void `glutInitDisplayMode`(unsigned int mode);
 - ▶ Specify a display mode for windows created
 - ▶ Color: `GLUT_RGBA`, `GLUT_RGB` or `GLUT_INDEX`
 - ▶ Framebuffer: `GLUT_SINGLE` or `GLUT_DOUBLE`
 - ▶ Buffer: `GLUT_DEPTH`, `GLUT_STENCIL` and `GLUT_ACCUM`

Initialization and window

- ▶ void `glutInitWindowSize`(int width, int height);
 - ▶ Set the initial window size
- ▶ void `glutInitWindowPosition`(int x, int y);
 - ▶ Set the initial window position
 - ▶ The actual position is left to the window system to determine
- ▶ int `glutCreateWindow`(char *name);
 - ▶ Create and open a window with previous settings

Initialization and window

- ▶ void `glutPostRedisplay`(void);
 - ▶ Mark the current window as needing to be redisplayed
 - ▶ The window's display callback will be called
- ▶ void `glutSwapBuffers`(void);
 - ▶ Swap the buffers of the current window if double buffered
 - ▶ An implicit `glFlush` is done by `glutSwapBuffers`

Callback Registration

- ▶ void `glutDisplayFunc`(void (*func)(void));
 - ▶ Put whatever you want to render in the callback
 - ▶ The callback is called when the window need to be redisplayed
 - ▶ Call `glutPostRedisplay()` to trigger the callback
- ▶ void `glutReshapeFunc`(void (*func)(int width, int height));
 - ▶ The callback is called when a window is created, resized or moved
 - ▶ Always call `glViewport()` to resize your viewport
- ▶ void `glutIdleFunc`(void (*func)(void));
 - ▶ Perform background processing tasks or continuous animation when window system events are not being received
 - ▶ The idle callback is continuously called when events are not being received

Callback Registration

- ▶ void `glutKeyboardFunc`(void (*func)(unsigned char key, int x, int y));
 - ▶ Each key press generating a keyboard callback
 - ▶ **key:** The ASCII character generated by the pressed key
 - ▶ x and y: The mouse location in window relative coordinates when the key was pressed
- ▶ void `glutMouseFunc`(void (*func)(int button, int state, int x, int y));
 - ▶ Each press and each release mouse button in a window generates a mouse callback
 - ▶ button: GLUT_LEFT_BUTTON, GLUT_MIDDLE_BUTTON or GLUT_RIGHT_BUTTON
 - ▶ state: GLUT_UP or GLUT_DOWN
 - ▶ x and y: The mouse location in window relative coordinates when the mouse button state changed

Callback Registration

- ▶ `void glutMotionFunc(void (*func)(int x, int y));`
 - ▶ The callback is called when the mouse moves within the window **while any mouse buttons are pressed**
 - ▶ x and y: the mouse location in window relative coordinates
- ▶ `void glutPassiveMotionFunc(void (*func)(int x, int y));`
 - ▶ The callback is called when the mouse moves within the window **while no mouse buttons are pressed**
 - ▶ x and y: the mouse location in window relative coordinates

Geometric Object Rendering

- ▶ void `glutSolidSphere`(GLdouble size); void `glutWireSphere`(GLdouble size);
- ▶ void `glutSolidCube`(GLdouble size); void `glutWireCube`(GLdouble size);
- ▶ void `glutSolidCone`(GLdouble size); void `glutWireCone`(GLdouble size);
- ▶ void `glutSolidTorus`(GLdouble size); void `glutWireTorus`(GLdouble size);
- ▶ void `glutSolidDodecahedron`(GLdouble size); void `glutWireDodecahedron`(GLdouble size);
- ▶ void `glutSolidOctahedron`(GLdouble size); void `glutWireOctahedron`(GLdouble size);
- ▶ void `glutSolidTetrahedron`(GLdouble size); void `glutWireTetrahedron`(GLdouble size);
- ▶ void `glutSolidIcosahedron`(GLdouble size); void `glutWireIcosahedron`(GLdouble size);
- ▶ void `glutSolidTeapot`(GLdouble size); void `glutWireTeapot`(GLdouble size);



Beginning Event Processing

- ▶ void `glutMainLoop(void);`
 - ▶ Enter the GLUT event processing loop
 - ▶ Once called, this routine will never return