Computer Graphics Programming

CGT 520

Computer Graphics Technology Dept.
Purdue University

January 20, 2015

Outline

The purpose

Since OpenGL is designed to be platform independent, it does **not** contain any window creation or window management functions.

Using GLUT, the same code can open windows under Microsoft, Linux, Macintosh and other operating systems and on a wide variety of hardware.

GLUT is easy to use, but there are other portable libraries that can be used with OpenGL. (e.g. SDL, Qt)

Types of GLUT functions

GLUT functions:

- Initialization of GLUT itself
- Window creation and management : OS independent
- Callback function registration : event driven

GLUT is **event-driven**. We will register callback functions at startup and GLUT will call the functions as needed.

All functions are named as glut*

Functions

void glutInit(int *argcp, char **argv);

Must be called before the window is created. Command line options are system dependent.

void glutInitDisplayMode(unsigned int mode);

Configures the frame buffer, as determined by mode. The flags specified by mode can include color, depth, stencil buffer, and single/double buffering modes.

Others:

- glutInitWindowPosition
- glutInitWindowSize

More on double buffering

The command

```
glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGBA);
```

will allow you to create 2 color buffers.

- Front buffer : the buffer currently viewable.
- Back buffer : the buffer currently drawable.
- Note how the mode is selected by logically 'or'ing flags together.

This prevents 'tearing' artifacts during animation caused by writing to the viewable buffer.

Functions

int glutCreateWindow(char *name);

Creates the window, name is displayed in the title bar.

void glutPostRedisplay(void);

Sets a flag indicating that the current window needs to be redrawn.

void glutSwapBuffers(void);

Swap front / back buffers when double buffering is enabled.(Usually implemented by just swapping pointers, not by copying buffer contents.)

Also, commands for setting / getting current window in applications with multiple windows.

Callback function registration

These functions set the callback function for the current window.

void glutDisplayFunc(void (*func)(void));

The function pointed to by func will be called from within the event loop when the redisplay flag is set. The callback function typically contains the OpenGL drawing commands.

void glutIdleFunc(void (*func)(void));

The function pointed to by func will be called from within the event loop while windowing system events are not being handled. The callback function typically contains code which performs some animation (but not rendering).

Call glutPostRedisplay at the end of the callback function so that the results of the animation will be displayed.

Callback function registration

void glutKeyboardFunc(void (*func)(unsigned char key, int x, int y));

The function pointed to by func will be called from within the event loop when an ASCII key is pressed and the window is current.

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

The function pointed to by func will be called from within the event loop when a mouse button is pressed or released and the window is current.

See the online resources and the glut.h header file for more details.

The event loop

void glutMainLoop(void);

This is event loop from which the registered callbacks are made. This function does not return*.

* newer versions of freeglut have glutLeaveMainLoop()

Typical program structure

- Initialize GLUT : glutInit* functions
- Create window : glutCreateWindow
- Register callbacks : glut*Func functions
- Initialize OpenGL state : Set color buffer clear color, camera parameters
- Enter event loop : glutMainLoop

- In idle callback: perform animation, call glutPostRedisplay
- In display callback: clear screen, draw scene, call glutSwapBuffers (if double-buffered)