来源： http://blog.sina.com.cn/s/blog_5f0d72800100iajl.html

penGL 中常用的 GLUT 函数库

(2010-05-01 19:33:44)

转载

标签：

opengl

分类： OpenGL

教育

GLUT 函数说明

一、初始化

void glutInit(int* argc,char** argv)

这个函数用来初始化 GLUT 库。对应 main 函数的形式应是： int main(int argc,char* argv[]);

这个函数从 main 函数获取其两个参数。

void glutInitWindowSize(int width,int height);
void glutInitWindowPosition(int x,int y);

设置 glut 程序要产生的窗口的大小和位置（左上角）。以像素为单位。

void glutInitDisplayMode(unsigned int mode);

设置图形显示模式。参数 mode 的可选值为：

- GLUT_RGBA：当未指明 GLUT-RGBA 或 GLUT-INDEX 时，是默认使用的模式。表明欲建立 RGBA 模式的窗口。
- GLUT_RGB：与 GLUT-RGBA 作用相同。
- GLUT_INDEX：指明为颜色索引模式。
- GLUT_SINGLE：只使用单缓存
- GLUT_DOUBLE：使用双缓存。以避免把计算机作图的过程都表现出来，或者为了平滑地实现动画。
- GLUT_ACCUM：让窗口使用累加的缓存。
- GLUT_ALPHA：让颜色缓冲区使用 alpha 组件。
- GLUT_DEPTH：使用深度缓存。
- GLUT_STENCIL：使用模板缓存。
- GLUT_MULTISAMPLE：让窗口支持多例程。
- GLUT_STEREO：使窗口支持立体。
- GLUT_LUMINACE:luminance 是亮度的意思。但是很遗憾，在多数 OpenGL 平台上，不被支持。

二、事件处理（Event Processing）

void glutMainLoop(void)

让 glut 程序进入事件循环。在一个 glut 程序中最多只能调用一次。一旦调用，会直到程序结束才返回。

三、窗口管理（Window Management）

int glutCreateWindow(char* name);

产生一个顶层的窗口。name 作为窗口的名字，也就是窗口标题栏显示的内容。

返回值是生成窗口的标记符，可用函数 glutGetWindow()加以引用。

int glutCreateSubWindow(int win,int x,int y,int width,int height);

创建一个子窗口。win 是其父窗口的标记符。x,y 是相对父窗口的位移，以像素表示。

width,height 是子窗口的宽和高。

void glutSetWindow(int win);
int glutGetWindow(void);

功能分别是：设置标记符为 win 的窗口为当前窗口；返回当前窗口的标记符。

void glutDestroyWindow(int win);

销毁以 win 标记的窗口。

void glutPostRedisplay(void);

将当前窗口打上标记，标记其需要再次显示。

void glutSwapBuffers(void);

当窗口模式为双缓存时，此函数的功能就是把后台缓存的内容交换到前台显示。当然，只有单缓存时，使用它的功能跟用 glFlush()一样。

而使用双缓存是为了把完整图画一次性显示在窗口上，或者是为了实现动画。

void glutPositionWindow(int x,int y);

改变当前窗口的位置：当前窗口是顶层窗口时，x,y 是相对于屏幕的的位移；当前窗口若是子窗口时，x,y 是相对其父窗口原点的位移。

void glutReshapeWindow(int width,int height);

改变当前窗口的大小。

width,height 是当前窗口新的宽度和高度值，当然只能是正值。

void glutFullscreen(void);

让当前窗口全屏显示。当前窗口是顶层窗口时才有效。

void glutPopWindow(void);
void glutPushWindow(void);

对顶层窗口和子窗口均有效。改变当前窗口在栈中相对于其它窗口的次序。

void glutShowWindow(void);
void glutHideWindow(void);
void glutIconifyWindow(void);

这三个函数作用是改变当前窗口的显示状态。

glutShowWindow 让当前窗口可视（这时它还是可能被其它窗口挡住）。
glutHideWindow 让当前窗口成为不可视状态。
glutIconifyWindow 让当前窗口成为一个图标，也即是最小化。

void glutSetWindowTitle(char* name);
void glutSetIconTitle(char* name);

设置当前窗口（必须是顶层窗口）的标题和图标化时的标题。

void glutSetCursor(int cursor);

设置当前窗口的光标样式。

cursor 可选值有许多：如 GLUT_CURSOR_RIGHT_ARROW 指向右边的光标，
GLUT_CURSOR_LEFT_ARROW 指向左边的光标，GLUT_CURSOR_INFO 成为手状。
GLUT_CURSOR_DESTROY 呈叉状，GLUT_CURSOR_HELP 呈现问号的形状。等等。

四、窗口的覆盖管理

void glutEstablishOverlay(void);

对当前窗口创建覆盖图层。该覆盖图的模式由初始化显示模式函数 glutDisplayMode()决定。

glutLayerGet(GLUT_OVERLAY_POSSIBLE)可用以设置对于当前窗口，是否允许产生由初始化显示模式函数规定其模式的覆盖图层。

void glutUserLayer(GLenum layer);

枚举量 layer 可选值为：GLUT_NORMAL,GLUT_OVERLAY.分别选取正常位平面或覆盖平面。

void glutRemoveLayer(void);

除去覆盖图。当没有覆盖图层时，调用这条语句也是安全的，这时系统不做任何事。

void glutPostOverlayRedisplay(void);

标记该覆盖图层为需要重新显示的状态。

void glutShowOverlay(void);
void glutHideOverlay(void);

显示当前窗口的覆盖图层；隐藏覆盖图层。这两条语句即时执行。注意一下，只有窗口可视时，使用 glutShowOverlay 才能使其覆盖图层可视。当窗口被其他窗口遮挡时，其覆盖图层也被遮挡从而不可视。

五、菜单管理

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

当点击菜单时,调用回调函数 func，value 为传递给回调函数的数值,它由所选择的菜单条目对应的整数值所决定。

这个函数创建一个新的弹出式菜单,并返回一个唯一的标识次菜单的整型标识符,并将新建的弹出菜单与 func 函数关联在一起,这样,当选择此菜单中的一个菜单条目时,调用回调函数 func.

```
void glutSetMenu(int menu);
int glutGetMenu(void);
```

设置当前菜单;获取当前菜单的标识符

```
void glutDestroyMenu(int menu);
```

删除指定的菜单

```
void glutAddMenuEntry(char* name, int value);
```

添加一个菜单条目

```
void glutAddSubMenu(char* name, int menu);
```

在当前菜单的底部增加一个子菜单的触发条目

```
void glutChangeToMenuEntry(int entry, char* name, int value);
```

更改当前菜单中指定菜单项

```
void glutChangeToSubMenu(int entry, char* name, int menu);
```

将指定的当前菜单中菜单项变为子菜单触发条目

```
void glutRemoveMenuItem(int entry);
```

删除指定的菜单项

```
void glutAttachMenu(int button);
void glutDetachMenu(int button);
```

把当前窗口的一个鼠标按键与当前菜单关联起来;解除鼠标按键与弹出式菜单的关联关系。

六、注册回调

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

为当前窗口设置显示回调函数

```
void glutOverlayDisplayFunc(void (*func)(void) );
```

注册当前窗口的重叠层的显示回调函数

```
void glutReshapeFunc(void (*Func)(int width, int height) );
```

指定当窗口的大小改变时调用的函数

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

注册当前窗口的键盘回调函数

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

注册当前窗口的鼠标回调函数

func 为注册的鼠标回调函数,这个函数完成鼠标事件的处理

button 为鼠标的按键,为以下定义的常量:

GLUT_LEFT_BUTTON	鼠标左键
GLUT_MIDDLE_BUTTON	鼠标中键
GLUT_RIGHT_BUTTON	鼠标右键

state 为鼠标按键的动作,为以下定义的常量:

GLUT_UP	鼠标释放
GLUT_DOWN	鼠标按下

x,y 为鼠标按下式,光标相对于窗口左上角的位置

```
void glutMotionFunc(void (*func)(int x, int y));
void glutPassiveMotionFunc(void (*func)(int x, int y));
```

设置移动回调函数;设置当前鼠标移动函数

Func 为注册的鼠标移动函数

x,y 为鼠标按下式,光标相对于窗口左上角的位置

当鼠标在窗口中按下并移动时调用 glutMotionFunc 注册的回调函数

当鼠标在窗口中移动时调用 glutPassiveMotionFunc 注册的回调函数

```
void glutVisibilityFunc(void (*func) (int state) );
```

设置当前窗口的可视回调函数

Func 为指定的可视回调函数

state 表示窗口的可视性,为以下常量:

GLUT_NOT_VISIBLE	窗口完全不可见
GLUT_VISIBLE	窗口可见或部分可见

这个函数设置当前窗口的可视回调函数,当窗口的可视性改变时,该窗口的可视回调函数被调用.只要窗口中的任何一个像素是可见的,或者他的任意一个子窗口中任意一个像素是可见的,GLUT 则认为窗口是可见的.

```
void glutEntryFunc(void (*func) (int state));
```

设置鼠标的进出窗口的回调函数

Func 为注册的鼠标进出回调函数

state 为鼠标的进出状态,为以下常量之一:

GLUT_LEFT	鼠标离开窗口
GLUT_RIGHT	鼠标进入窗口

当窗口取得焦点或失去焦点时调用这个函数,当鼠标进入窗口区域并点击时,state 为 GLUT_RIGHT,当鼠标离开窗口区域点击其他窗口时,state 为 GLUT_LEFT.

void glutSpecialFunc(void (*func) (int key, int x, int y))

设置当前窗口的特定键的回调函数

Func 为注册的特定键的回调函数

key 为按下的特定键,为以下定义的常量:

key 常量	描述
GLUT_KEY_F1	F1 功能键
GLUT_KEY_F2	F2 功能键
GLUT_KEY_F3	F3 功能键
GLUT_KEY_F4	F4 功能键
GLUT_KEY_F5	F5 功能键
GLUT_KEY_F6	F6 功能键
GLUT_KEY_F7	F7 功能键
GLUT_KEY_F8	F8 功能键
GLUT_KEY_F9	F9 功能键
GLUT_KEY_F10	F10 功能键
GLUT_KEY_F11	F11 功能键
GLUT_KEY_F12	F12 功能键
GLUT_KEY_LEFT	左方向键
GLUT_KEY_UP	上方向键
GLUT_KEY_RIGHT	右方向键
GLUT_KEY_DOWN	下方向键
GLUT_KEY_PAGE_UP	PageUp 键
GLUT_KEY_PAGE_DOWN	PageDown 键
GLUT_KEY_HOME	Home 键
GLUT_KEY_END	End 键

GLUT_KEY_INSERT	Insert 键

x,y 为当按下键时鼠标的坐标,相对于窗口左上角,以像素为单位

注意:ESC,回车和 delete 键由 ASCII 码产生.

void glutMenuStatusFunc(void (*func) (int status, int x, int y));

设置菜单状态回调函数

func 是注册的菜单状态回调函数

status 是当前是否使用菜单,为以下定义的常量:

GLUT_MENU_IN_USE	菜单正在使用
GLUT_MENU_NOT_IN_USE	菜单未被使用

x,y 是鼠标按下式,光标相对于窗口左上角的位置

这个函数时 glut 程序判定是否正在使用菜单,当弹出菜单时,调用注册的菜单状态回调函数,同时 status 设置为常量 GLUT_MENU_IN_USE, 当菜单使用完毕时,也调用菜单状态回调函数,此时 status 变量变为 GLUT_MENU_NOT_IN_USE.从已弹出的菜单中再弹出的菜单不产生菜单状态回调过程.每个 glut 程序只有一个菜单状态回调函数.

glutSpaceballRotateFunc

glutSpaceballButtonFunc

glutButtonBoxFunc

glutDialsFunc

glutTabletMotionFunc

glutTabletButtonFunc

void glutMenuStatusFunc(void (*func) (int status, int x, int y));

设置菜单状态回调函数

func 为注册的菜单状态回调函数

status 表示当前是否使用菜单,为以下定义的常量:

GLUT_MENU_IN_USE	菜单正在使用
GLUT_MENU_NOT_IN_USE	菜单未被使用

x,y 表示鼠标按下式,光标相对于窗口左上角的位置

这个函数时 glut 程序判定是否正在使用菜单,当弹出菜单时,调用注册的菜单状态回调函数,同时 status 设置为常量 GLUT_MENU_IN_USE, 当菜单使用完毕时,也调用菜单状态回调函数,此时 status 变量变为 GLUT_MENU_NOT_IN_USE.从已弹出的菜单中再弹出的菜单不产生菜单状态回调过程.每个 glut 程序只有一个菜单状态回调函数.

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

设置空闲回调函数

func 表示当系统空闲时调用的函数,它的形式为 void func(void)

void glutTimerFunc(unsigned int msecs, void (*Func)(int value), int value);

注册一个回调函数,当指定时间值到达后,由 GLUT 调用注册的函数一次

msecs 是等待的时间

Func 是注册的函数

value 是指定的一个数值,用来传递到回调函数 Func 中

这个函数注册了一个回调函数,当指定的毫秒数到达后,这个函数就调用注册的函数,value 参数用来向这个注册的函数中传递参数。

七、色彩管理(未完成)

glutSetColor

glutGetColor

glutCopyColormap

八、状态检索

int glutGet(GLenum state);

检索指定的 GLUT 状态

state 为指定要检索的状态类型,为以下常量:

state 常量	描述
GLUT_WINDOW_X	当前窗口的 x 坐标,以像素为单位
GLUT_WINDOW_Y	当前窗口的 y 坐标,以像素为单位
GLUT_WINDOW_WIDTH	当其窗口的宽度,以像素为单位
GLUT_WINDOW_HEIGHT	当前窗口的高度,以像素为单位
GLUT_WINDOW_BUFFER_SIZE	当前窗口中,颜色分量占用的位数,即用多少 bit 表示颜色分量
GLUT_WINDOW_STENCIL_SIZE	当前窗口中,蒙板分量占用的位数,即用多少 bit 表示蒙板分量
GLUT_WINDOW_DEPTH_SIZE	当前窗口中,深度分量占用的位数,即用多少 bit 表示深度分量
GLUT_WINDOW_RED_SIZE	当前窗口中,红色分量占用的位数,即用多少 bit 表示红色分量
GLUT_WINDOW_GREEN_SIZE	当前窗口中,绿色分量占用的位数,即用多少 bit 表示绿色分量
GLUT_WINDOW_BLUE_SIZE	当前窗口中,蓝色分量占用的位数,即用多少 bit 表示蓝色分量
GLUT_WINDOW_ALPHA_SIZE	当前窗口中,alpha 色分量占用的位数,即用多少 bit 表示 alpha 色分量
GLUT_WINDOW_ACCUM_RED_SIZE	当前窗口累积缓存中,红色分量占用的位数,即用多少 bit 表示红色分量
GLUT_WINDOW_ACCUM_GREEN_SIZE	当前窗口累积缓存中,绿色分量占用的位数,即用多少 bit 表示绿色分量

GLUT_WINDOW_ACCUM_BLUE_SIZE 当前窗口累积缓存中,蓝色分量占用的位数,即用多少 bit 表示蓝色分量
GLUT_WINDOW_ACCUM_ALPHA_SIZE 当前窗口累积缓存中,alpha 色分量占用的位数,即用多少 bit 表示 alpha 色分量
GLUT_WINDOW_DOUBLEBUFFER 如果窗口式双缓存模式,返回 1,否则返回 0
GLUT_WINDOW_RGBA 如果窗口是 RGBA 模式,返回 1,否则返回 0
GLUT_WINDOW_PARENT 查询当前窗口的父窗口个数,如果为顶层窗口返回 0
GLUT_WINDOW_NUM_CHILDREN 查询当前窗口的子窗口个数
GLUT_WINDOW_NUM_SAMPLES 查询多重采样的采样点个数
GLUT_WINDOW_STEREO 查询是否使用立体模式,是则返回 1,否则返回 0
GLUT_WINDOW_CURSOR 返回光标的整数标示
GLUT_SCREEN_HEIGHT 屏幕的高度,以像素为单位
GLUT_SCREEN_WIDTH 屏幕的宽度,以像素为单位
GLUT_SCREEN_WIDTH_MM 屏幕的宽度,以毫米为单位
GLUT_SCREEN_HEIGHT_MM 屏幕的高度,以毫米为单位
GLUT_MENU_NUM_ITEMS 查询当前菜单包含的菜单项的个数
GLUT_DISPLAY_MODE_POSSIBLE 查询窗口系统是否支持当前的显示模式,1 表示支持,0 表示不支持
GLUT_INIT_DISPLAY_MODE 初始窗口的显示模式
GLUT_INIT_WINDOW_X 初始窗口的 x 坐标
GLUT_INIT_WINDOW_Y 初始窗口的 y 坐标
GLUT_INIT_WINDOW_WIDTH 初始窗口的宽度
GLUT_INIT_WINDOW_HEIGHT 初始窗口的高度
GLUT_ELAPSED_TIME 返回两次调用 glutGet(GLUT_ELAPSED_TIME)的时间间隔,单位为毫秒

返回值根据查询的内容返回相应的值,无效的状态名返回-1.
int glutLayerGet(GLenum info);
查询属于当前窗口的重叠层的状态
Info 表示查询的重叠层状态常量:

GLUT_OVERLAY_POSSIBLE	在给定的初始显示模式下,能否为当前窗口创建重叠层.如果能,返回 1;如果不能,返回 0
GLUT_LAYER_IN_USE	返回当前的使用层,为 GLUT_NORMAL 或 GLUT_OVERLAY
GLUT_HAS_OVERLAY	判断当前窗口是否创建了重叠层
GLUT_NORMAL_DAMAGED	如果当前窗口的图像层在上一次显示回调函数调用后已经破坏,则返回 TRUE
GLUT_OVERLAY_DAMAGED	如果当前窗口的重叠层在上一次显示回调函数调用后已经破坏,则返回 TRUE

int glutDeviceGet(GLenum info);
检索设备信息
info 为要检索的设备信息的名字,为以下常量:

GLUT_HAS_KEYBOARD	如果键盘可用,返回非 0 值,否则,返回 0
GLUT_HAS_MOUSE	如果鼠标可用,返回非 0 值,否则,返回 0
GLUT_NUM_MOUSE_BUTTONS	返回鼠标支持的按键数,如果鼠标不可用,返回 0

返回值 0 表示检索的设备不存在,非 0 表示设备可用
int glutGetModifiers(void);
返回组合功能键的状态
返回值为以下定义的常量:

GLUT_ACTIVE_SHIFT	当按下 shift 键时
GLUT_ACTIVE_CTRL	当按下 ctrl 键时
GLUT_ACTIVE_ALT	当按下 alt 键时

int glutExtensionSupported(char* extension);
判定是否支持特定的 OpenGL 扩展。
extension 是指定要测试的 OpenGL 扩展的名称
如果给定扩展获得支持,函数返回非 0,否则返回 0。

九、实体绘制

以下所有函数中，radius 表示球体的半径，slices 表示球体围绕 z 轴分割的数目（经线），stacks 表示球体沿着 z 轴分割的数目（纬线）。
绘制中心在模型坐标原点,半径为 radius 的球体,球体围绕 z 轴分割 slices 次,球体沿着 z 轴分割 stacks 次
void glutWireSphere(GLdouble radius, GLint slices, GLint stacks); 线框球
void glutSolidSphere(GLdouble radius, GLint slices, GLint stacks); 实心球
void glutWireCube(GLdouble size); 线框立方体
void glutSolidCube(GLdouble size); 实心立方体
void glutWireTorus(GLdouble innerRadius, GLdouble outerRadius, GLint nsides, GLint rings); 线框圆环
void glutSolidTorus(GLdouble innerRadius, GLdouble outerRadius, GLint nsides, GLint rings); 实心圆环
void glutWireIcosahedron(void); 线框 20 面体
void glutSolidIcosahedron(void); 实心 20 面体
void glutWireOctahedron(void); 线框 8 面体
void glutSolidOctahedron(void); 实心 8 面体
void glutWireTetrahedron(void); 线框 4 面体
void glutSolidTetrahedron(void); 实心 4 面体

void glutWireDodecahedron(GLdouble radius); 线框12面体
void glutSolidDodecahedron(GLdouble radius); 实心12面体
void glutWireCone(GLdouble radius, GLdouble height, GLint slices, GLint stacks); 线框圆锥体
void glutSolidCone(GLdouble radius, GLdouble height, GLint slices, GLint stacks); 实心圆锥体
void glutWireTeapot(GLdouble size); 线框茶壶
void glutSolidTeapot(GLdouble size); 实心茶壶