Keyboard Mouse and Menus

Reshape Callback

- Whenever a window is initialized, moved or resized, the window sends an event to notify us of the change
- When we use GLUT, the event will be handled by the function registered by

glutReshapeFunc( )

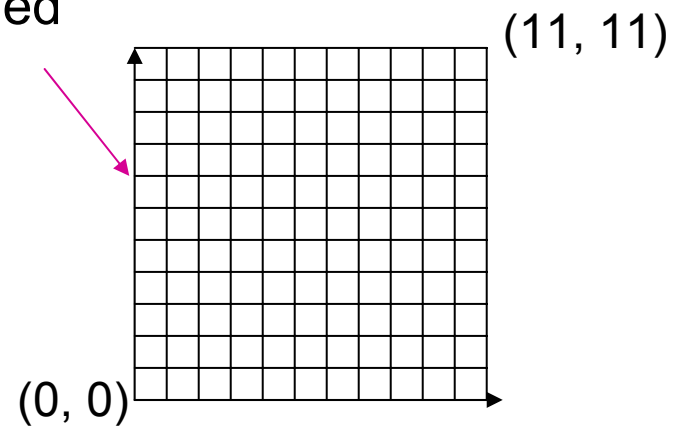
- This callback function should:
  - Reestablish the rectangular-region that will be the new rendering-area
  - Define the coordinate-system to which objects will be drawn

Reshape Callback

Example:

```
void reshape(int w, int h)
{
    glViewPort(0, 0, (GLIsizei) w, (GLIsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, (GLsizei) w, 0.0, (GLsizei) h);
}
```

The coordinate system defined by w = 11 and h = 11:



Hidden Surface Removal

- OpenGL uses the depth-buffer method (also called Z-buffer)
- The method works by associating a depth for EACH pixel

Z-buffer Algorithm:

- Depth values for all pixels are set to the largest possible value
- Draw objects in the scene in ANY order. Before each pixel is drawn, a comparison is done with the depth-value already stored at the pixel. The new pixel is drawn only if its z coordinate is smaller than the Z value in the buffer
- The Z-buffer is eventually updated with the distance of the new pixel

Pseudocode:

```
if (new_pixel_closer_than(prev_pixel_z) {
    draw_new_pixel( );
    store_new_z(pixel_z);
}
```

Hidden Surface Removal

Example:

```
glutInitDisplayMode(GLUT_DEPTH | ….);
glEnable(GL_DEPTH_TEST);
   :
while (1) {
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
   draw_3d_object_A( );
   draw_3d_object_B( );
}
```

Keyboard Callbacks

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

Registers the function f( ) that is called when a key is pressed; (x, y) is always measured from top left corner of the window

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

Registers the callback that handles special keys (function, arrows, etc…)

int glutGetModifiers( )

Return one of GLUT_ACTIVE_SHIFT, GLUT_ACTIVE_CTRL or GLUT_ACTIVE_ALT if the key is pressed when a keyboard or mouse event are generated

Keyboard Callbacks

Example:

The callback registered by glutSpecialFunc( ) will have tests like:

```
if(key == GLUT_KEY_F1) …. // Function key F1
if(key == GLUT_KEY_F2) …. // Function key F2
```

The constants associated to the special keys are defined in glut.h

Keyboard Callbacks

```
#define GLUT_KEY_F1                     1
#define GLUT_KEY_F2                     2
#define GLUT_KEY_F3                     3
#define GLUT_KEY_F4                     4
#define GLUT_KEY_F5                     5
#define GLUT_KEY_F6                     6
#define GLUT_KEY_F7                     7
#define GLUT_KEY_F8                     8
#define GLUT_KEY_F9                     9
#define GLUT_KEY_F10                    10
#define GLUT_KEY_F11                    11
#define GLUT_KEY_F12                    12
/* directional keys */
#define GLUT_KEY_LEFT                   100
#define GLUT_KEY_UP                     101
#define GLUT_KEY_RIGHT                  102
#define GLUT_KEY_DOWN                   103
#define GLUT_KEY_PAGE_UP                104
#define GLUT_KEY_PAGE_DOWN              105
#define GLUT_KEY_HOME                   106
#define GLUT_KEY_END                    107
#define GLUT_KEY_INSERT                 108
```

Keyboard Callbacks

Example:

To have Control-c or Control-C terminate a program

```
if ((glutGetModifiers( ) == GLUT_ACTIVE_CTRL) &&
        ((key == ‘c’) || (key == ‘C’))) exit(0)
```

NOTE: The modifiers also work with mouse functions

Mouse Callback

void glutMouseFunc(void (*f)(int button, int state, int x, int y))

Registers the mouse callback f( ) which handles the position (x, y) of the mouse in the window, its state (GLUT_UP or GLUT_DOWN) and the button causing the event (GLUT_LEFT_BUTTON, GLUT_RIGHT_BUTTON GLUT_MIDDLE_BUTTON)

EXAMPLE:

```
void mymouse(int button, int state, int x, int y) {
    if(state == GLUT_DOWN &&
       button == GLUT_LEFT_BUTTON)
    exit( );
}
```

Mouse Motion

```
void  glutMotionFunc(void (*f)(int x, int y))
void  glutPassiveMotionFunc(void (*f)(int x, int y))
```

Specify motion and passive motion (no button pressed) callback functions

```
void  glutEntryFunc(void (*f)(int state))
```

State is GLUT_ENTERED or GLUT_LEFT depending on the mouse entering or leaving the window area

Menus

int glutCreateMenu (void (*f)(int value))

Creates top level menu that uses callback f( ) which is passed an integer value from the menu entry. Returns a unique menu identifier

void glutSetMenu (int id)

Sets the current menu to id

void glutAddMenuEntry (char *name, int value)

Adds entry to the current menu. Name is the entry name and value is returned to the menu callback

Menus

int glutAttachMenu (int button)

Attaches current menu to the specified mouse button. Button is GLUT_LEFT_BUTTON, GLUT_RIGHT_BUTTON or GLUT_MIDDLE_BUTTON

void glutAddSubMenu (char *name, int menu)

Adds a submenu entry name as the next entry in the current menu. The value of menu is the id of the submenu returned when the submenu was created

Menus

Example:

```
glutCreateMenu(mymenu);  // single menu, no need for id
glutAddMenuEntry("Clear Screen", 1);
glutAddMenuEntry("Exit", 2);
glutAttachMenu(GLUT_RIGHT_BUTTON);
```

Callback:

```
void mymenu(int value) {
    if(value == 1)
        glClear( );
    if(value == 2)
        exit(0);
}
```