

The User Interface

Lecture 2 Sections 2.11

Robb T. Koether

Hampden-Sydney College

Fri, Aug 26, 2011

Outline

- 1 GLUT Callback Functions
- 2 Coordinating the Callback Functions
- 3 The `keyboard()` Function
- 4 The `mouse()` Function
- 5 Assignment

Outline

- 1 GLUT Callback Functions
- 2 Coordinating the Callback Functions
- 3 The `keyboard()` Function
- 4 The `mouse()` Function
- 5 Assignment

Callback Functions

Definition (Callback Function)

A **callback function** is a programmer-specified function that the library will call whenever necessary.

- Each callback function must be registered with GLUT.
- GLUT provides over 20 callbacks.

Callback Functions

- The GLUT library contains functions with names of the form

```
glutNameFunc ( function ) ,
```

where *Name* stands for some form of windows interaction (mouse, keyboard, etc.) and *function* is the name of the function that handles that interaction.

- For example,

```
glutMouseFunc (mouse) ;
```

- Then when the action is initiated by the user (mouse click, keystroke, etc.), the function is called to handle it.

The Mouse Function

```
glutMouseFunc (mouse) ;
```

- Called whenever the mouse is clicked (up or down).
- Activated by mouse clicks.
- Left or right button, button up or button down.

The Keyboard Function

```
glutKeyboardFunc (keyboard) ;
```

- Called whenever an ASCII key is pressed.
- Activated by keystrokes (down only) of an ASCII key (letters, digits, punctuation).

The Special Function

```
glutSpecialFunc (special);
```

- Called whenever a non-ASCII key is pressed.
- Activated by keystrokes (down only) of a non-ASCII key (function keys, arrow keys, etc.).

The Motion Function

```
glutMotionFunc (motion) ;
```

- Called whenever the mouse is moved *while the mouse button is down*.

The Passive Motion Function

```
glutPassiveMotionFunc (passiveMotion) ;
```

- Called whenever the mouse is moved *while the mouse button is up*.

The Display Function

```
glutDisplayFunc (display);
```

- Called whenever the scene needs to be redrawn.
- Activated by calls to `glutPostRedisplay()`.

The Reshape Function

```
glutReshapeFunc (reshape) ;
```

- Called whenever the window is resized.
- Activated by resizing the window.
- Includes a call to `glutPostRedisplay()`.

The Idle Function

```
glutIdleFunc (idle);
```

- Called whenever nothing else is happening.

Outline

- 1 GLUT Callback Functions
- 2 Coordinating the Callback Functions**
- 3 The `keyboard()` Function
- 4 The `mouse()` Function
- 5 Assignment

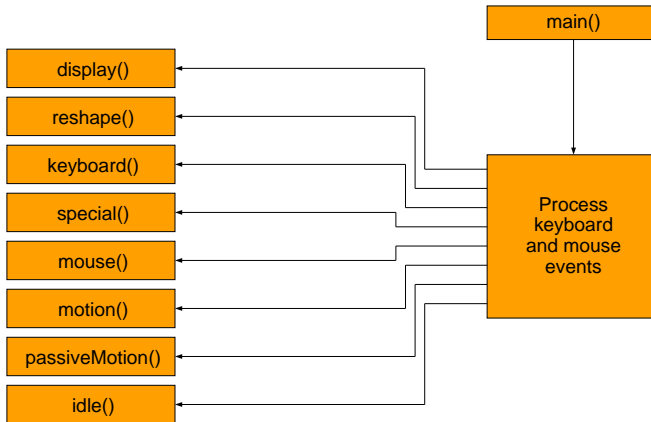
OpenGL Callback Functions

- If the scene needs to be redrawn, then the callback functions should end with a call to `glutPostRedisplay()`.
- To force the scene to be redrawn continuously, include `glutPostRedisplay()` in the `display()` function or in the `idle()` function.

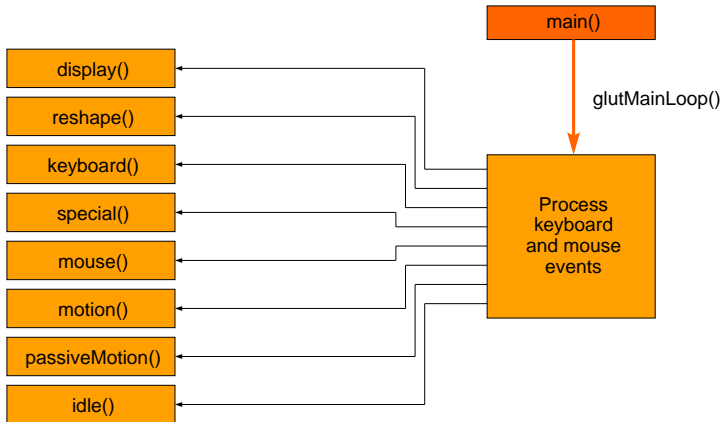
The Main Loop

- `main()` ends by calling `glutMainLoop()`.
- This function runs “forever,” or until we exit the program.
- It calls the callback functions as necessary.
- It handles all drawing commands as they are generated.

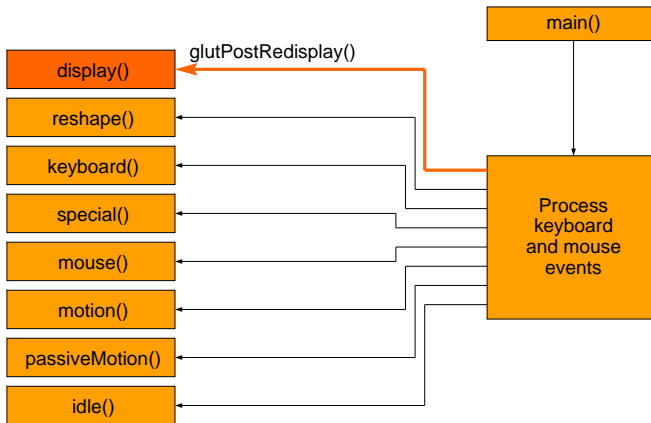
The Main Loop



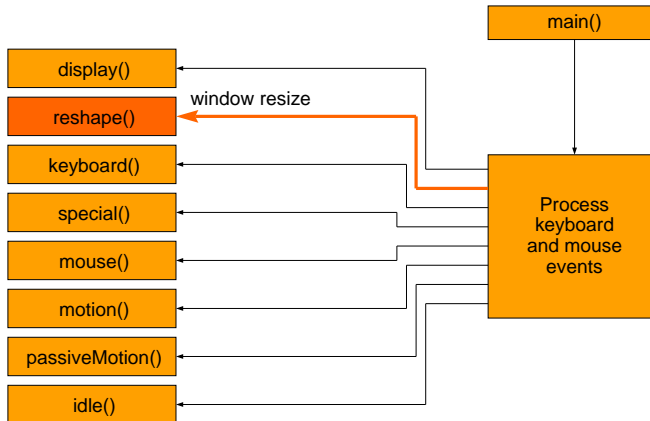
The Main Loop



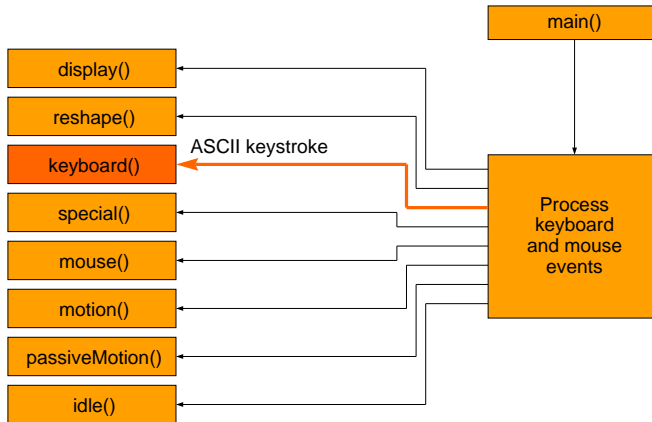
The Main Loop



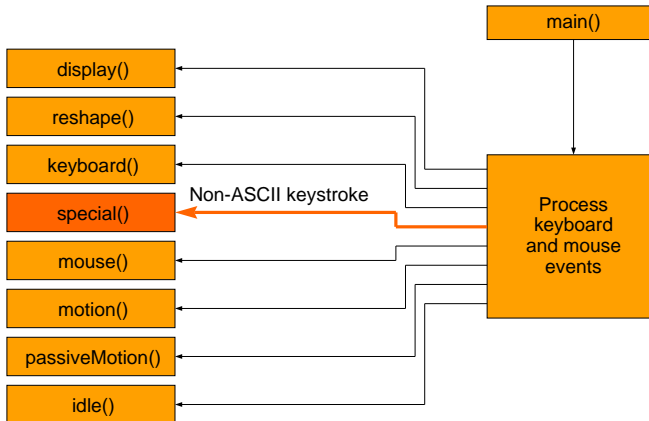
The Main Loop



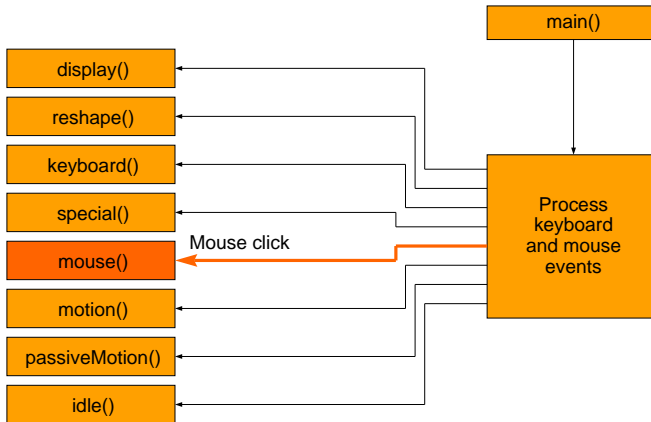
The Main Loop



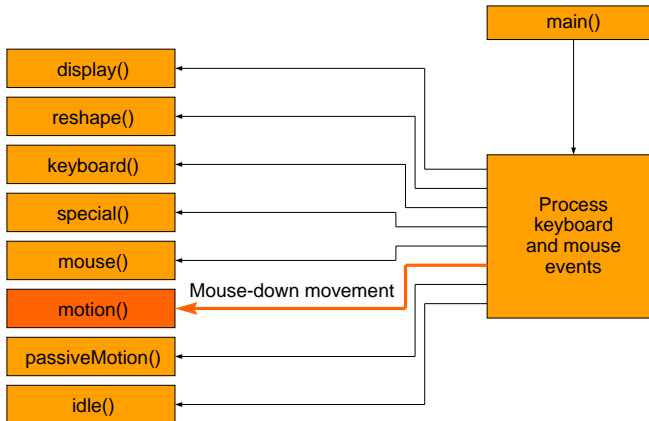
The Main Loop



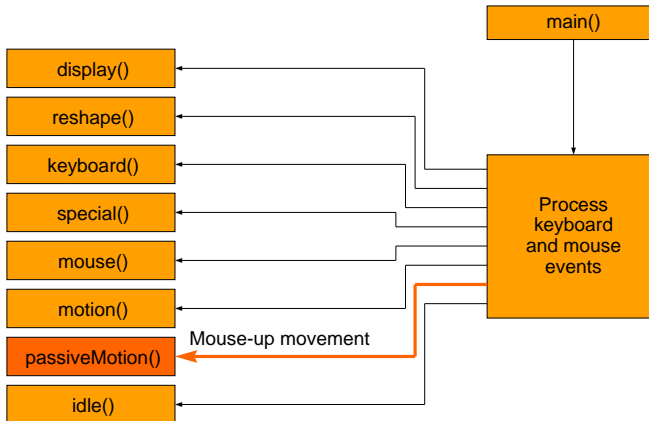
The Main Loop



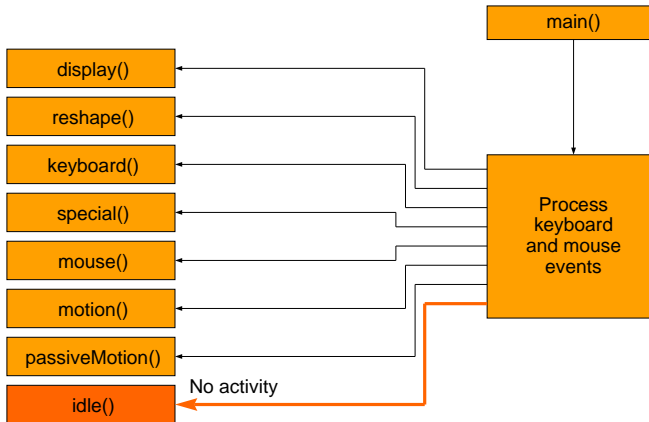
The Main Loop



The Main Loop



The Main Loop



Callback Functions

Example (Callback Functions)

- The code.
- The executable.

Callback Functions

Example (Callback Functions)

- The code.
- The executable.

Other Initializations

Other Initializations

```
void init ()
{
    glClearColor(0.8, 0.8, 0.8, 0.0);
    glEnable(GL_DEPTH_TEST);
    glDepthFunc(GL_LEQUAL);
    glClearDepth(1.0);
    printInstructions();
    :
return;
}
```

Outline

- 1 GLUT Callback Functions
- 2 Coordinating the Callback Functions
- 3 The `keyboard()` Function**
- 4 The `mouse()` Function
- 5 Assignment

The keyboard () Function

The keyboard () Function

```
void keyboard(unsigned char key, int x, int y);
```

- `key` – ASCII value of the key pressed.
- `x, y` – `x` and `y` window coordinates of the mouse.
- Caution – `y` is measured from the top down.

The keyboard () Function

Example (The keyboard () Function)

```
void keyboard(unsigned char key, int x, int y)
{
    y = screenHeight - y;
    switch (key)
    {
        case '+': case '=':
            // Code to zoom in
            break;
            :
        case ESC:
            exit(0);
            break;
    }
    glPostRedisplay();
    return;
}
```

Typical structure of the keyboard () function

Outline

- 1 GLUT Callback Functions
- 2 Coordinating the Callback Functions
- 3 The `keyboard()` Function
- 4 The `mouse()` Function**
- 5 Assignment

The `mouse()` Function

The `mouse()` Function

```
void mouse(int button, int state, int x, int y);
```

- `button` – `GLUT_LEFT_BUTTON` or `GLUT_RIGHT_BUTTON`.
- `state` – `GLUT_UP` or `GLUT_DOWN`.
- `x`, `y` – `x` and `y` window coordinates of mouse.
- Caution – `y` is measured from the top down.

The keyboard () Function

Example (The keyboard () Function)

```
void mouse(int button, int state, int x, int y)
{
    y = screenHeight - y;
    if (button == GLUT_LEFT_BUTTON
        && state == GLUT_DOWN)
    {
        : // Perform left-button-down action
    }
    else
        : // Other actions
    glPostRedisplay();
    return;
}
```

Typical structure of the mouse () function

Outline

- 1 GLUT Callback Functions
- 2 Coordinating the Callback Functions
- 3 The `keyboard()` Function
- 4 The `mouse()` Function
- 5 Assignment**

Homework

Homework

- Read Section 2.11 – the user-interface functions.
- Read Section 2.12 – creating pop-up menus.