The User Interface

Lecture 2
Sections 3.6, 3.7

Robby T. Koether

Hampden-Sydney College

Fri, Aug 28, 2009
Outline

1. GLUT Callback Functions
2. Coordinating the Callback Functions
3. The `keyboard()` Function
4. The `mouse()` Function
5. Assignment
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.
The Glut library contains functions with names of the form

\[ \text{glutNameFunc}(\text{function}) \]

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

For example,

\[ \text{glutMouseFunc(mouse)}; \]

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

```c
glutDisplayFunc(display);
```

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

Glut Callback Functions

Coordinating the Callback Functions

The Keyboard Function

The Mouse Function

Assignment
Glut Callback Functions

The Reshape Function

```c
glutReshapeFunc(reshape);
```

- Called whenever the window is resized.
- Activated by resizing the window.
Glut Callback Functions

The Keyboard Function

```c
glutKeyboardFunc(keyboard);
```

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

The User Interface
Robb T. Koether

GLUT Callback Functions

Coordinating the Callback Functions

The keyboard() Function

The mouse() Function

Assignment

The Special Function

```c
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 Mouse Function

```c
glutMouseFunc(mouse);
```

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

The Motion Function

```
 glutMotionFunc(motion);
```

- Called whenever the mouse is moved while the mouse button is down.
Glut Callback Functions

The Passive Motion Function

```c
glutPassiveMotionFunc(passageMotion);
```

*Called whenever the mouse is moved *while the mouse button is up.*
Glut Callback Functions

The Idle Function

```c
glutIdleFunc(idle);
```

- Called whenever nothing else is happening.
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.
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

main()
  →
  Process keyboard and mouse events

  ↓
  display()
  reshape()
  keyboard()
  special()
  mouse()
  motion()
  passiveMotion()
  idle()
The Main Loop

```
main()
glutMainLoop()
```

- `display()`
- `reshape()`
- `keyboard()`
- `special()`
- `mouse()`
- `motion()`
- `passiveMotion()`
- `idle()`
The Main Loop

The User Interface
Robb T. Koether

GLUT Callback Functions

Coordinating the Callback Functions

The keyboard() Function
The mouse() Function
Assignment

The Main Loop

- main()
  - Process keyboard and mouse events
    - display()
      - glutPostRedisplay()
    - reshape()
    - keyboard()
    - special()
    - mouse()
    - motion()
    - passiveMotion()
    - idle()
The Main Loop

- main()

  - process keyboard and mouse events

- display()
- reshape()
- keyboard()
- special()
- mouse()
- motion()
- passiveMotion()
- idle()

window resize
The Main Loop

- main()
- display()
- reshape()
- keyboard()
- special()
- mouse()
- motion()
- passiveMotion()
- idle()

Process keyboard and mouse events
The Main Loop

- main()
- display()
- reshape()
- keyboard()
- special()
- mouse()
- motion()
- passiveMotion()
- idle()
- Process keyboard and mouse events

Non-ASCII keystroke
The Main Loop

- `main()`
- `display()`
- `reshape()`
- `keyboard()`
- `special()`
- `mouse()`
- `motion()`
- `passiveMotion()`
- `idle()`

Process keyboard and mouse events
The Main Loop

main()

Process keyboard and mouse events

display()
reshape()
keyboard()
special()
mouse()
motion()
passiveMotion()
idle()

Mouse-down movement

The User Interface
Robb T. Koether

GLUT Callback Functions
Coordinating the Callback Functions
The keyboard() Function
The mouse() Function
Assignment
The Main Loop

main()
Process keyboard and mouse events

- display()
- reshape()
- keyboard()
- special()
- mouse()
- motion()
- passiveMotion()
- idle()

Mouse-up movement
The Main Loop

- display()
- reshape()
- keyboard()
- special()
- mouse()
- motion()
- passiveMotion()
- idle()

main()

Process keyboard and mouse events

No activity
Example (Callback Functions)

- The code.
- The executable.
Other Initializations

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

```c
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)

```c
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
The mouse() Function

```c
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)

```c
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
Homework

- Read Section 3.6 – the user-interface functions.
- Read Section 3.7 – creating pop-up menus.