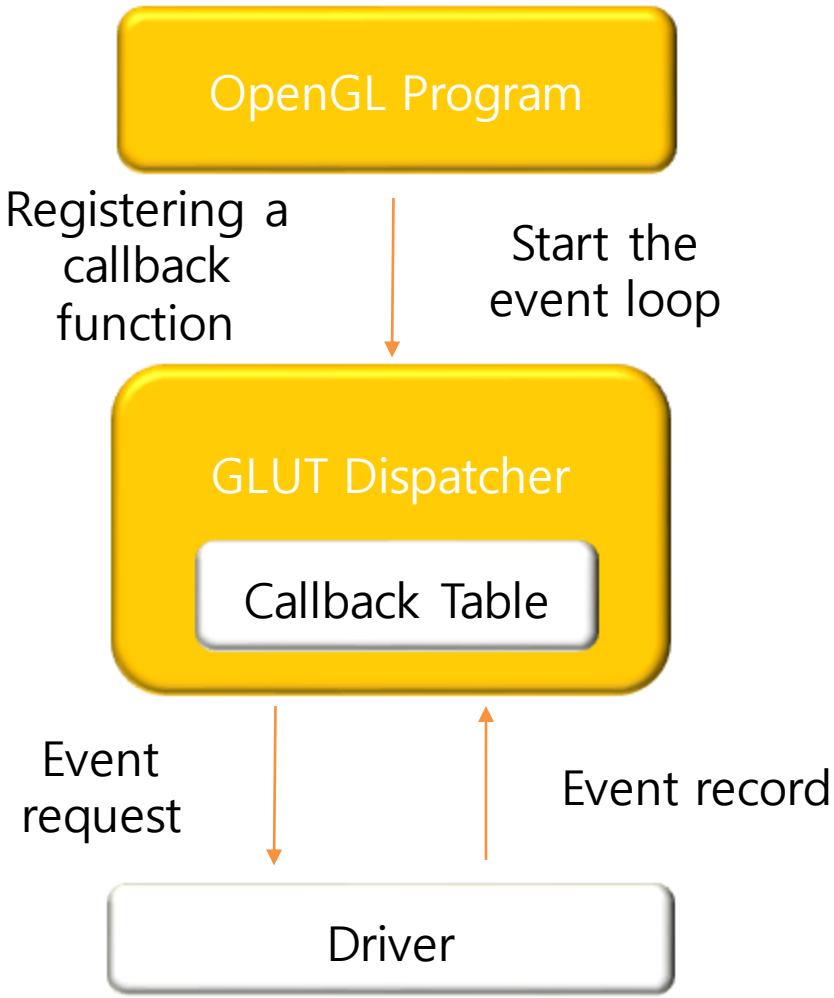
Computer Graphics
- [Open GL] Callback Functions

Sung Soo Hwang

Input Callback and GLUT



<Callback Table>

Event Type	Callback Functions (example)
DISPLAY	MyDisplay()
RESHAPE	MyReshape()
KEYBOARD	MyKeyBoard()
MOUSE	MyMouse()
IDLE	MyIdle()

<Type-specific callback functions>

Registering callback function by event type

Event Type	Callback Function Registration Command(Example)	Callback Function Prototype
DISPLAY	glutDIsplayFunc(MyDisplay)	void MyDisplay();
MOUSE	glutMouseFunc(MyMouse)	void MyMouse(int button, int state, int x, int y)
KEYBOARD	glutKeyboardFunc(MyKeyboard)	void MyKeyboard(char key, int x, int y)
RESHAPE	glutReshapeFunc(MyReshape)	void MyReshape(int width, int height)
IDLE	glutIdleFunc(MyIdle)	void MyIdle();

Reshape Callback

- GLUT treats the reshape event as occurring in the following three cases.

1) When opening a window for the first time
2) When moving the window position
3) When resizing the window

The callback function prototype for registration of Reshape event

=> `void glutReshapeFunc(void(*func)(int width, int height));`

Reshape Callback

- Example

... triangles.cpp ...

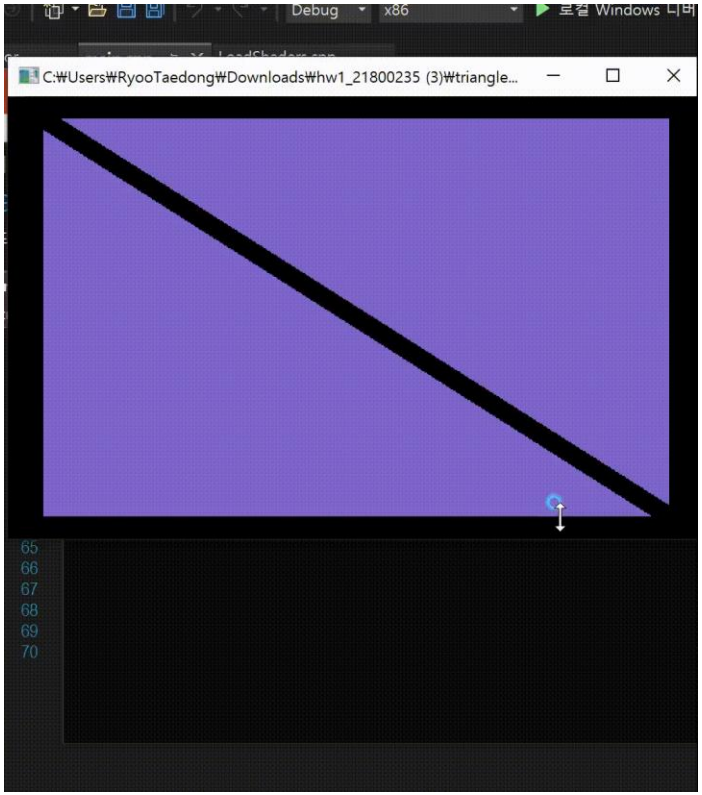
```
void main(int argc, char** argv)
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_RGBA);
        glutInitWindowSize(512, 512);
        glutCreateWindow(argv[0]);
        GLenum err = glewInit();
        if (err != GLEW_OK) {
                fprintf(stderr, "Error: %s₩n", glewGetErrorString(err));
                exit(EXIT_FAILURE);
        }
        init();

        glutDisplayFunc(display);
        glutReshapeFunc(reshape);
        glutMainLoop();
}
```

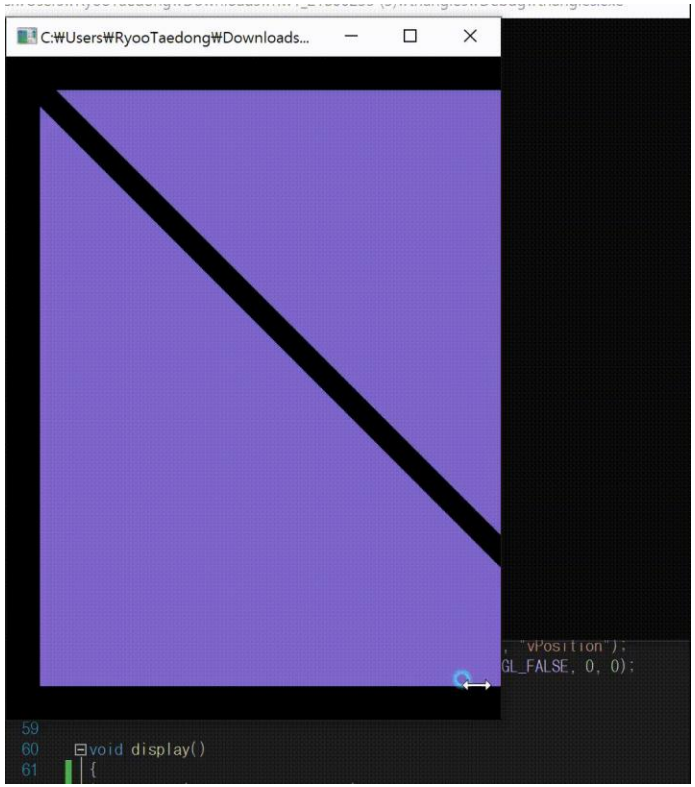
Reshape Callback

... triangles.cpp ...

```
void display()
{
        glClear(GL_COLOR_BUFFER_BIT);
        glBindVertexArray(VertexArrays[0]);
        glDrawArrays(GL_TRIANGLES, 0, NumVertices);
        glFlush();
}

void reshape(int w, int h) {
        glViewport(0, 0, w, h);
}
```

Reshape Callback






```
void reshape(int w, int h) {
        glViewport(0, 0, w, h);
}
```

```
void reshape(int w, int h) {
        //glViewport(0, 0, w, h);
}
```

Keyboard Callback

- Keyboard callbacks
  - void cb_keyboard(unsigned char key, int x, int y)
    - key: ASCII character of the pressed key
    - x, y: mouse position
    - It is registered by glutKeyboardFunc(cb_keyboard).

*ASCII: American Standard Code for Information Interchange

Keyboard Callback

- Keyboard callbacks
  - void cb_special(int key, int x, int y)
    - key: non-ASCII of the pressed key
    - x, y: mouse position
    - It is registered by glutSpecialFunc(cb_special).

GLUT_KEY_F1, GLUT_KEY_F2, ..., GLUT_KEY_F12	F1 through F12 keys
GLUT_KEY_PAGE_UP, GLUT_KEY_PAGE_DOWN	Page Up and Page Down keys
GLUT_KEY_HOME, GLUT_KEY_END	Home and End keys
GLUT_KEY_LEFT, GLUT_KEY_RIGHT, GLUT_KEY_UP, GLUT_KEY_DOWN	Arrow keys
GLUT_KEY_INSERT	Insert key

Keyboard Callback

- How to deal with shift, ctrl, and alt modifiers?

- int glutGetModifiers(void)
  - Returns the state of modifier keys (shift, ctrl, alt) at the time when the input event for a keyboard, special, or mouse callback is generated.
  - The return value is generated from the following constants:

GLUT_ACTIVE_SHIFT	Set if the Shift modifier is active
GLUT_ACTIVE_CTRL	Set if the Ctrl modifier is active
GLUT_ACTIVE_ALT	Set if the Alt modifier is active

  - Note : there can be multiple active modifiers, which can be checked with the bitwise AND operator (&) as follows:

```
int modifiers = glutGetModifiers();
if (modifiers & GLUT_ACTIVE_CTRL) printf("ctrl pressed₩n");
if (modifiers & GLUT_ACTIVE_ALT) printf("alt pressed₩n");
if (modifiers & GLUT_ACTIVE_SHIFT) printf("shift pressed₩n");
```

Keyboard Callback

- Example

... triangles.cpp ...

```
void main(int argc, char** argv)
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_RGBA);
        glutInitWindowSize(512, 512);
        glutCreateWindow(argv[0]);
        GLenum err = glewInit();
        if (err != GLEW_OK) {
                fprintf(stderr, "Error: %s₩n", glewGetErrorString(err));
                exit(EXIT_FAILURE);
        }
        init();

        glutKeyboardFunc(keyboard);
        glutMainLoop();
}
```

Keyboard Callback

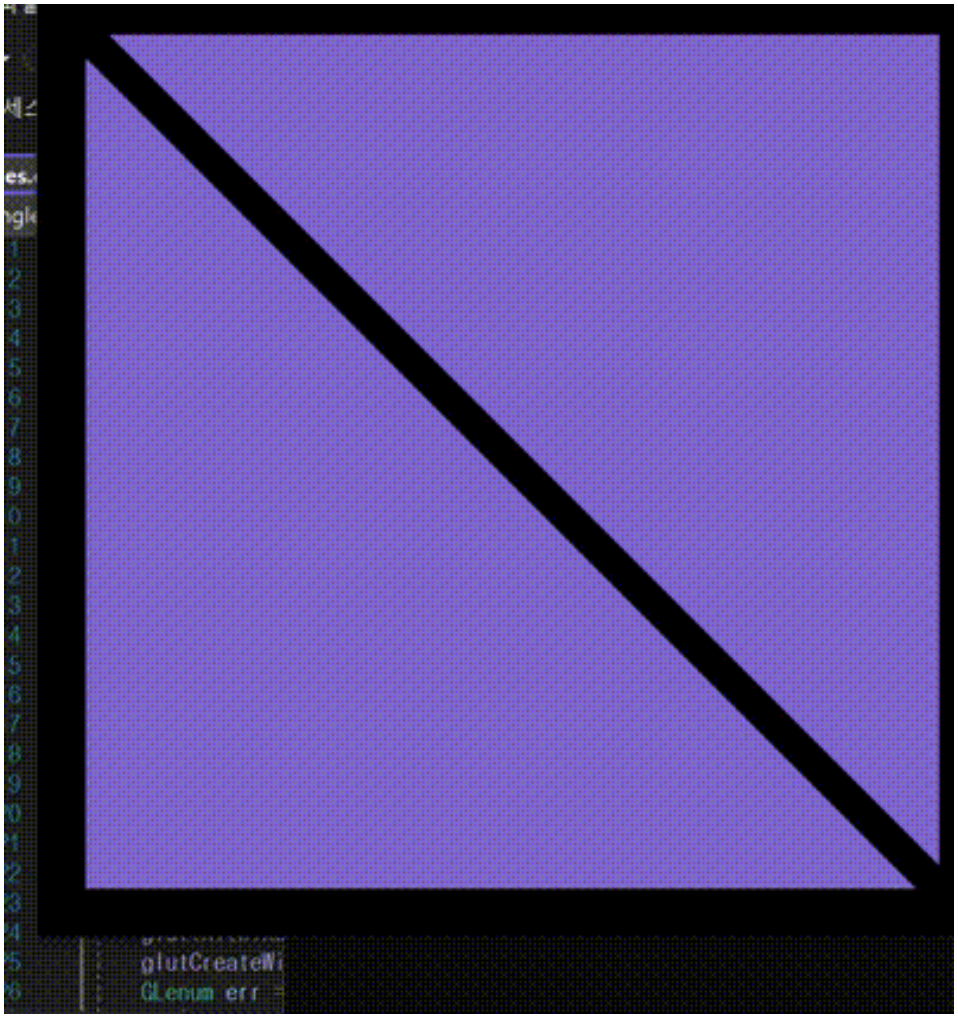
... triangles.cpp ...

```
void keyboard(unsigned char KeyPressed, int x, int y) {
        switch (KeyPressed) {

        case 'Q':
                exit(0); break;

        case 'q':
                exit(0); break;

        case 27:
                exit(0); break;
        }
}
```

ASCII ESC

Keyboard Callback

- Press ESC or q to end the program.



Mouse Callback

- Mouse Callbacks
  - void cb_mouse(int button, int state, int x, int y)
    - button: represents which button is selected
    - state: represents the button state
    - x, y: mouse position
    - It is registered by glutMouseFunc(cb_mouse).

Constants	Val.	Description
GLUT_LEFT_BUTTON GLUT_MIDDLE_BUTTON GLUT_RIGHT_BUTTON	0 1 2	Left Mouse Button Middle Mouse Button Right Mouse Button
GLUT_DOWN GLUT_UP	0 1	Button pressed Button released

Mouse Callback

- Mouse Callbacks
  - void cb_motion(int x, int y)
  - void cb_passive_motion(int x, int y)
    - Called when the mouse moves within the window while one or more mouse buttons are pressed(cb_motion) or while no mouse buttons are pressed (cb_passive_motion)
    - x, y: mouse position
    - They are registered by glutMotionFunc(cb_motion) and glutPassiveMotionFunc(cb_passive_motion), respectively

Mouse Callback

- Mouse Callbacks
  - void cb_entry(int state)
    - State: represents the entry status of the mouse pointer

GLUT_LEFT	The mouse pointer has left the window
GLUT_ENTERED	The mouse pointer has entered the window

    - It is registered by glutEntryFunc(cb_entry).

  - void cb_wheel(int wheel, int direction, int x, int y)
    - wheel: wheel number (which seems not well defined)
    - direction: +1 or -1 depending on the direction of the scroll
    - x, y: mouse position
    - It is registered by glutMouseWheelFunc(cb_wheel).