



R E P O R T

컴퓨터그래픽스및실습 실습과제 04

과목명	컴퓨터그래픽스및실습
분반	01
교수	최영규
학번	2020136129
이름	최수연
제출일	2022년 10월 3일 월요일

문제 분석 및 해결 방법

[문제 설명]

- 0.5초마다 다음과 같이 GLUT 3D 객체를 순서대로 출력하는 프로그램을 구현하라.
 - Cube → sphere → cone → ... → teapot → 자신의 이름 → Sierpinski Gasket → 반복
- 키보드 이벤트 추가
 - 'a': 자동 갱신 모드를 토글(toggle)한다. 즉, 이전에 ON이라면 OFF, OFF면 ON으로 변경됨.
자동갱신모드가 비활성화되면 더 이상 타이머 콜백이 자동으로 호출되지 않음.
 - '0'~'8': 그 번호에 해당하는 객체를 출력. 자동 갱신 모드는 해제되어야 함.
 - 'i': 객체의 회전이 초기화 됨.
 - 'q': 프로그램 종료.
- 마우스 이벤트 추가
 - 마우스의 왼쪽 버튼을 눌러서 드래그하면 그려지는 물체가 회전한다.
 - 마우스가 좌우 또는 상하로 움직이는 정도에 따라 물체는 y축 또는 x축을 따라 회전한다.
 - 이를 위해 마우스 클릭과 마우스 모션의 두 콜백함수를 사용해야 한다.
- Reshape 이벤트 추가
 - 윈도우 크기를 변경하더라도 출력되는 객체의 가로세로 비율은 똑같이 유지되도록 처리한다.
- 메뉴 처리
 - 팝업 메뉴를 만든다.
 - 메뉴에는 각 GLUT 객체를 출력한다는 메시지가 나타난다.
 - 각 메뉴를 누를 때마다 해당 기능이 실행된다.

[문제 분석 및 해결 방법]

- 0.5초마다 객체 출력되도록 하려면 glutTimerFunc() 함수를 사용하여 타이머를 설정해 화면이 전환되도록 구현해야 한다.
- 키보드 이벤트는 glutKeyboardFunc() 함수를 사용해 if 문을 통해 a, i, q, 0 ~ 8를 누를 때 위 조건에 맞게 변경되도록 구현한다.
- 마우스 이벤트는 glutMouseFunc()과 glutMotionFunc() 함수를 사용하여 마우스 클릭과 마우스 모션 이벤트를 처리한다.
- Reshape 이벤트는 glutReshapeFunc() 함수를 사용해 reshape 함수를 구현하여 윈도우 창을 늘리거나 위치를 변경해도 왜곡이 되지 않도록 구현한다.
- 팝업 메뉴를 만들기 위해 glutCreateMenu()와 glutAddMenuEntry() 함수를 사용하여 각 객체를 메뉴별로 연결하여 메뉴 클릭 시 해당 객체로 화면 전환될 수 있도록 한다.

주요 설명 코드

```
static int objNum = 0; // 타이머나 키보드 입력값에 따라 변경되는 정적 변수
static bool bAuto = true; // 키보드 'a'가 눌렸을 때 ON/OFF 상태를 확인하기 위한 변수

void display() { // switch 문을 사용하여 objnum의 숫자가 변경될 때마다 해당 기능이 실행되도록 함
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 0.0, 0.0);
    switch (objNum) {
        case 0: glutWireCube(1); break;
        case 1: glutWireSphere(0.8, 16, 10); break;
        case 2: glutWireCone(0.8, 1, 20, 10); break;
        case 3: glutWireTorus(0.1, 0.8, 10, 20); break;
        case 4: glutWireTetrahedron(); break;
        case 5: glutWireIcosahedron(); break;
        case 6: glutWireTeapot(0.6); break;
        case 7: MyNameDisplay(); break; // 자신의 이름
        case 8: SierpinskiDisplay(); break; // Sierpinski Gasket
    }
    glFlush();
}

void myTimer(int Value) { // 타이머 함수
    glutPostRedisplay(); // 윈도우 화면이 변경될 때, 윈도우를 재생하도록 요청하는 함수
    if (bAuto) { // bAuto가 true일 때, objNum에 1을 더하고 0.5초 간격으로 화면에 도형 출력
        objNum = (objNum + 1) % 9;
        glutTimerFunc(500, myTimer, 1);
    }
}

void keyboard(unsigned char key, int x, int y) { // 키보드 함수
    if (key >= '0' && key <= '8') { // 키보드가 0 ~ 8 사이의 숫자가 눌렸을 때 objNum 설정
        objNum = key - '0';
        bAuto = false; // 숫자가 눌리면 bAuto를 false로 전환하여 화면이 계속 멈춰있도록 설정
    }
    else if (key == 'i') { // 키보드 i가 눌리면
        glMatrixMode(GL_MODELVIEW); // 실제 위치 지정
        glLoadIdentity(); // 단위 행렬로 초기화
    }
    else if (key == 'a') { // 키보드 a가 눌리면 bAuto를 반대로 바꾸고, 만약 true면 타이머 함수 실행
        bAuto = !bAuto;
        if (bAuto) glutTimerFunc(500, myTimer, 1);
    }
    else if (key == 'q') exit(0); // 키보드 q를 누르면 종료
    glutPostRedisplay();
}
```

```

}

void reshape(int width, int height) { // 윈도우 크기 변경 시에도 객체 비율 고정을 위한 함수
    glViewport(0, 0, width, height); // 뷰포트 설정
    double wScale = (double)width / 500; // 너비 설정
    double hScale = (double)height / 500; // 높이 설정
    glMatrixMode(GL_PROJECTION); // 화면 위치 계산 및 설정
    glLoadIdentity(); // 단위 행렬로 초기화
    glOrtho(-wScale, wScale, -hScale, hScale, -1, 1); // 육면체의 가시부피 설정
}

static int PrevX, PrevY; // 정적 변수 x, y 선언

void mouseClicked(int button, int state, int x, int y) { // 마우스 클릭 함수
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) { // 마우스 왼쪽 버튼이 눌린 상태면,
        PrevX = x; // 변수에 x 좌표 저장
        PrevY = y; // 변수에 y 좌표 저장
    }
}

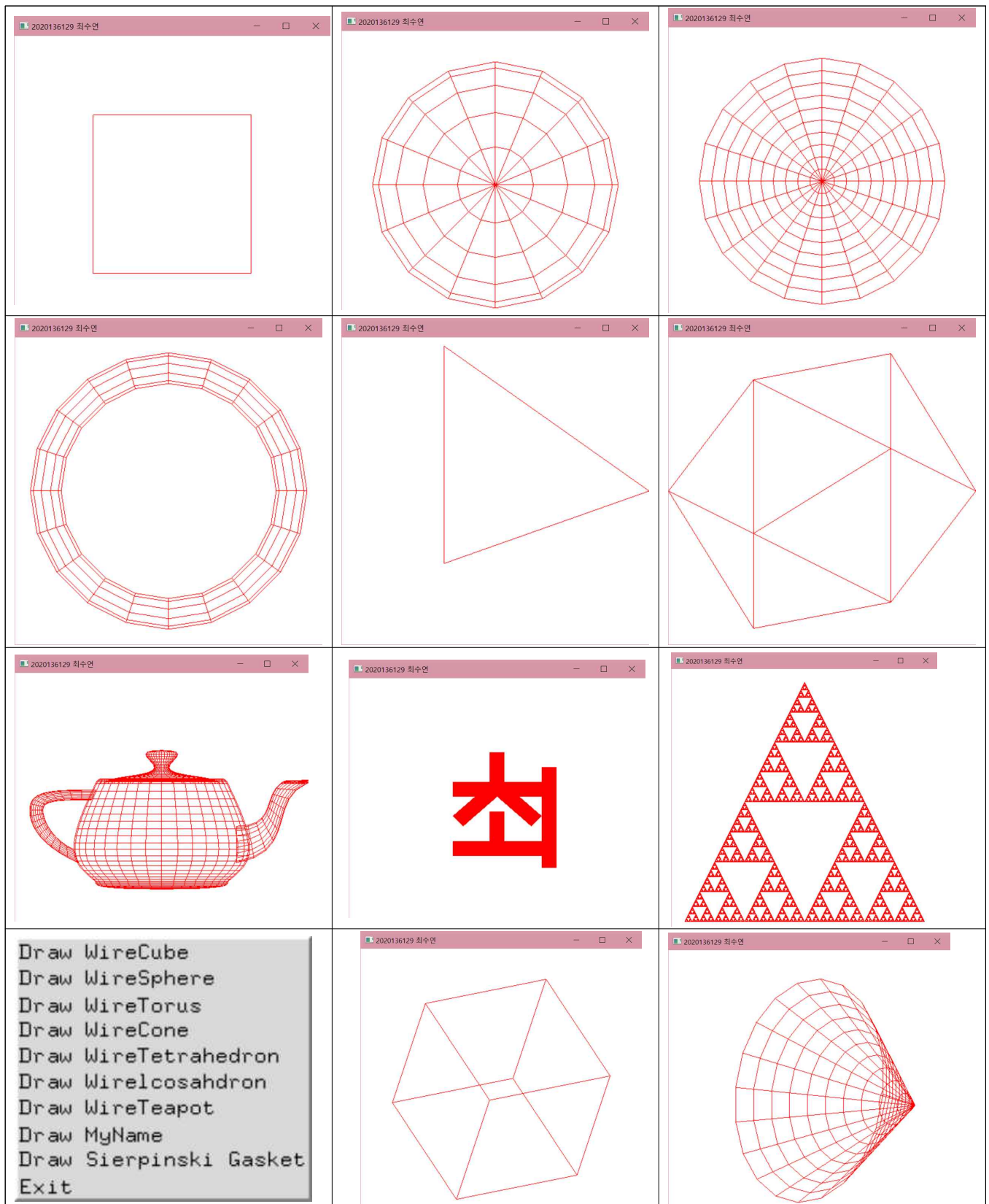
void mouseMotion(GLint x, GLint y) { // 마우스 모션 함수
    glMatrixMode(GL_MODELVIEW); // 실제 위치 지정
    glRotated(x - PrevX, 0, 1, 0); // 마우스 모션 함수를 통해 움직인 x 좌표만큼 행렬 계산 후 회전
    glRotated(y - PrevY, 1, 0, 0); // 마우스 모션 함수를 통해 움직인 y 좌표만큼 행렬 계산 후 회전
    PrevX = x; // 변수에 x 좌표 저장
    PrevY = y; // 변수에 y 좌표 저장
    glutPostRedisplay();
}

void myMenu(int id) { // initMenu에서 id를 인수로 전달받아 keyboard 함수로 전달, '9'번이 눌리면 종료.
    if (id >= 0 && id <= 8)
        keyboard(id + '0', 0, 0);
    else if (id == 9) exit(0);
}

void initMenu() { // 위에서 설정한 '0' ~ '9'번에 맞게 메뉴 등록
    GLint MyMainMenuID = glutCreateMenu(myMenu); // myMenu 함수로 인수 전달
    glutAddMenuEntry("Draw WireCube", 0);
    glutAddMenuEntry("Draw WireSphere", 1);
    glutAddMenuEntry("Draw WireTorus", 2);
    glutAddMenuEntry("Draw WireCone", 3);
    glutAddMenuEntry("Draw WireTetrahedron", 4);
    glutAddMenuEntry("Draw WireIcosahedron", 5);
    glutAddMenuEntry("Draw WireTeapot", 6);
    glutAddMenuEntry("Draw MyName", 7);
    glutAddMenuEntry("Draw Sierpinski Gasket", 8);
    glutAddMenuEntry("Exit", 9);
    glutAttachMenu(GLUT_RIGHT_BUTTON); // 우클릭 시 팝업 메뉴 호출
}

```

테스트 결과



고찰 및 느낀점

윈도우 화면으로 다양한 기능을 수행할 수 있어서 신기했다. 마우스 자체로 화면을 움직이거나 팝업 메뉴를 띄울 수 있는 것도 이미 glut에 다 내장된 콜백함수 덕분에 가능한 것인데, 이런 라이브러리를 구현할 수 있다는 것이 대단하다고 생각했다. 또, 수업 시간에 이론적으로 설명해주실 때는 잘 이해가 가지 않던 함수들을 직접 사용해보니까 어떻게 사용하는 건지 감도 잡히고 이벤트가 눈에 바로 보이니까 재밌었다.