**기초 컴퓨터 그래픽스**

**HW1 README**

**1. [환경 명세]**

window 10 64bit, Intel(R) Core(TM) i7-1065G7 CPU, visual studio 2022 – win 64, release

**2. [요구사항]**

모든 요구사항을 구현했다. 다각형 완성 여부는 iscomplete 전역 변수에 저장되고, 다각형 회전 상태는 isrotate 전역 변수에 저장된다.

**(a) 윈도우 화면**

- 작동 확인 방법: 실행하면 바로 뜬다.

- 구현 방법: glutInitWindowSize 함수를 사용하여 윈도우를 설정하였다.

**(b) 점 선택하기**

- 작동 확인 방법: shift키를 누르고 왼쪽 마우스 버튼을 클릭한다. (shift키를 누르지 않은 상태에서는 작동하지 않는다.)

- 기타 요구사항: **(g)** 다각형을 완성한 상태에서는 반응하지 않는다. **(m)** 다각형이 회전하고 있는 상태에서는 반응하지 않는다.

- 구현 방법: mousepress 함수에서 구현했다. iscomplete 변수를 통해 다각형이 완성되지 않음을 확인하고, glutGetModifiers 함수를 통해 shift키가 눌려진 상태인지 확인한다. (다각형이 완성되지 않으면 회전도 하지 않으므로 isrotate 변수는 확인하지 않는다.) 두 조건을 모두 만족하면 vertexx, vertexy 배열에 마우스의 좌표를 openGL좌표로 변환해서 저장한다. 이후 화면을 갱신하다.

display 함수에서 glBegin(GL\_POINTS)를 통해 vertexx, vertexy 배열에 저장된 점을 그린다.

**(c) 선분 그리기**

- 작동 확인 방법: 두 번째 점을 그릴 때부터, 자동으로 이전 점과 현재 그린 점 사이에 선분이 그려진다.

- 구현 방법: display 함수에서 glBegin(GL\_LINE\_STRIP)을 통해 vertexx, vertexy 배열에 저장된 점을 이은 선분들을 그린다.

**(d) 다각형 완성하기**

- 작동 확인 방법: p키를 누른다.

- 기타 요구사항: **(f)** 최소 3개의 점을 선택하지 않은 상태일 때, 콘솔에 에러 메시지를 출력한다.

- 구현 방법: keyboard 함수에서 p키를 인식하면 다음 조건에 따라 행동을 달리한다. 점의 개수(vnum)가 2보다 많으면, iscomplete 전역 변수 값을 1로 하고 화면을 갱신한다. 해당 조건을 만족하지 않으면 fprintf 함수를 통해 콘솔에 에러 메시지를 출력한다.

display 함수에서 glBegin(GL\_LINE\_LOOP)을 통해 vertexx, vertexy에 저장된 점을 연결해서 다각형을 그린다.

**(e) 점과 선분들 또는 다각형 지우기**

- 작동 확인 방법: c키를 누른다.

- 기타 요구사항: **(m)** 다각형이 회전하고 있으면 작동하지 않는다.

- 구현 방법: keyboard 함수에서 c키를 인식하면 점의 개수(vnum) 값을 0으로 초기화한다. 다각형 완성 여부와 회전 여부를 저장하는 iscomplete, isrotate 값을 0으로 초기화한다. 이후 화면을 갱신한다.

이때, 아무런 점을 선택하지 않았거나, 다각형이 회전하고 있으면 작동하지 않는다.

**(h) 다각형 이동하기1**

- 작동 확인 방법: 화살표 키를 누른다.

- 기타 요구사항: **(m)** 다각형이 회전하고 있으면 작동하지 않는다.

- 구현 방법: special 함수에서 왼쪽 화살표 키를 인식하면 vertexx 배열의 모든 원소에서 SENSITIVITY(0.01) 만큼 뺀다. 오른쪽 화살표 키를 인식하면 vertexx 배열의 모든 원소에서 SENSITIVITY 만큼 더한다. 아래쪽 화살표 키를 인식하면 vertexy 배열의 모든 원소에서 SENSITIVITY 만큼 뺀다. 위쪽 화살표 키를 인식하면 vertexy 배열의 모든 원소에서 SENSITIVITY 만큼 더한다. 이후 화면을 갱신한다.

이때 isrotate=1 이거나 iscomplete=0이면 작동하지 않는다.

**(i) 다각형 이동하기2**

- 작동 확인 방법: 오른쪽 마우스 버튼을 클릭한 상태에서 커서를 움직인다.

- 기타 요구사항: **(m)** 다각형이 회전하고 있으면 작동하지 않는다.

- 구현 방법: mousepress 함수에서 오른쪽 버튼을 누른 상태를 인식하고, isrotate=0, iscomplete=1이면 현재 커서 위치를 tmpx, tmpy에 저장한다. rightbuttonpressed 값을 1로 한다.

mousemove 함수에서 rightbuttonpressed 변수를 통해 오른쪽 마우스 버튼이 눌러진 상태임을 인식하면, 현재 커서 위치와 과거 커서 위치(tmpx, tmpy) 사이 거리를 구해서 vertexx,vertexy 배열에 저장된 좌표값에 더해준다. 현재 커서 위치를 tmpx, tmpy에 저장하고 화면을 갱신한다.

오른쪽 마우스 버튼이 눌러지지 않은 상태이면 rightbuttonpressed 값을 0으로 한다.

**(j) 다각형 회전하기**

- 작동 확인 방법: r키를 누른다.

- 구현 방법: keyboard 함수에서 r키를 인식하고, iscomplete 변수를 통해 다각형이 완성됐는지 확인한다. isrotate 변수를 통해 다각형이 회전하고 있지 않은 상태인지 확인한다. find\_centroid() 함수에서 무게 중심의 좌표를 계산하고, centerx, centery 변수에 저장한다. isrotate 값을 1로 설정하고 glutTimerFunc 함수를 통해 100ms 마다 rotate 함수를 실행한다.

rotate 함수는 isrotate=1이면 vertexx, vertexy 의 값을 RAD(0.05) 만큼 회전시킨다. 그 후 화면을 갱신하고, 100ms 후에 자기 자신을 다시 호출한다.

**(k) 다각형 무게 중심 그리기**

- 작동 확인 방법: r키를 누른다.

- 구현 방법: display 함수에서 isrotate=1 이면, glBegin(GL\_POINTS)로 좌표가 (centerx, centery)인 점을 그린다.

**(l) 다각형 회전 멈추고, 화면에서 무게 중심 사라지게 하기**

- 작동 확인 방법: 다각형이 회전하고 있을 때, r키를 누른다.

- 구현 방법: keyboard 함수에서 r키를 인식하고, 다각형이 회전하고 있는 상태이면, isrotate 값을 0으로 바꾼다.

**(n) 윈도우 크기를 변경하면 가로-세로 비율을 유지하면서 윈도우의 내용도 변경한다.**

- 작동 확인 방법: 마우스로 윈도우 크기를 변경하면 자동으로 내용도 바뀐다.

- 구현 방법: reshape 함수에서 glViewport 함수를 통해 뷰포트 크기를 바뀐 화면 크기로 설정한다.

**(o) 프로그램을 종료한다.**

- 작동 확인 방법: f키를 누른다.

- 구현 방법: keyboard 함수에서 f키를 인식하면, glutLeaveMainLoop 함수를 통해 main에서 돌고 있던 루프를 탈출한다.

**(p) 윈도우 배경, 점, 선분, 다각형의 무게 중심은 서로 다른 색깔로 그린다.**

- glClearColor 함수를 통해 윈도우 배경을 하얀색으로 설정했다 glColor3f 함수를 통해 점은 검은색, 선분은 빨간색, 무게 중심은 초록색으로 그려지게 설정했다.