

# 기초 컴퓨터 그래픽스 HW-1 README

학번 : 20171612  
이름 : 김성일

## 환경 명세

- Windows 10 64 bit
- Intel(R) Core(TM) i5-8250U CPU
- Intel(R) UHD Graphics 620
- Visual Studio 22
- win64, release

## 요구사항

| 모든 요구 사항을 구현하였기 때문에, 구현 여부를 생략하였습니다.

## 0. 클래스 설계

- Point 클래스
  - 좌표를 저장하는 클래스
  - 필드
    - x : x 좌표
    - y : y좌표
  - 메소드
    - 단순 x, y를 return하는 getter

- windowHeight, windowHeight에 따른 상대적인 좌표값을 return하는 getter
- PointManager 클래스
  - 내부에 vector<Point>를 필드로 갖는 vector의 역할을 위임받아 Point를 관리하는 클래스
  - 필드
    - vector<Point> points : 현재까지 입력받은 좌표 값
  - 메소드
    - getSize : 좌표의 개수 return
    - moveLeft, Right, Up, Down : 모든 좌표 값을 적절한 값 ( 5.0 ) 만큼 증가시키는 함수
    - updateRatio(ratioX, ratioY) : ratioX, ratioY 만큼 모든 좌표 값을 상대적인 크기만큼 변경
    - move(dx,dy) : dx, dy 만큼 모든 좌표 값을 이동
    - getAllPoints() : 모든 좌표를 갖는 vector를 return
    - getPoint(int offset) : vector내의 특정 offset의 좌표를 return
    - addPoint(Point point) : vector에 새로운 좌표를 추가.
    - getCenterPoint() : 무게중심 공식을 통해 vector내의 좌표를 기반으로 무게중심 계산

## 1. 윈도우 창 생성

- 작동 확인 방법 : 실행시 바로 화면이 동작
- 구현 방법
  - init 함수에서 WindowSize 및 WindowPosition을 정의하고 glutCreateWindow를 통해 화면을 생성.
  - glutDisplayFunc 에 정의한 display 함수를 파라미터로 넘겨주었고, display 함수에서 배경색 초기화 및 점 그리기, 선 그리기 등을 해주게 구현하였다.

## 2. 점 그리기 및 선 그리기 ( b, c )

- 작동 확인 방법 : SHIFT + 왼쪽 클릭

- 구현 방법

- `if ((button == GLUT_LEFT_BUTTON) && (state == GLUT_DOWN) && (glutGetModifiers() == GLUT_ACTIVE_SHIFT))` 를 통해 SHIFT + 왼쪽 클릭을 감지
  - 감지후, 완성된 Polygon이 아니면 PointManager라는 클래스에 현재 찍힌 점을 추가
- display 함수를 통해 화면 구성
  - drawDot 및 drawLine 함수에서 windowWidth, windowHeight를 통해 점 좌표를 다시 계산
    - x 좌표 : `(float)(x) / (float)(windowWidth / 2) - 1.0f`
    - y 좌표 : `((float)(y) / (float)(windowHeight / 2) - 1.0f)`
  - 계산된 좌표에 점을 찍거나 이전 좌표와 현재 좌표 사이를 이어주는 선을 생성

### 3. 다각형 완성하기 ( d, f )

- 작동 확인 방법 : p 를 눌러 확인
- 구현 방법
  - PointManager 클래스에서 getSize를 통해 현재 점의 개수 파악
    - 현재 점의 개수가 3개 이상이면 isComplete 라는 flag를 true로 변경
      - display 함수에서 isComplete를 체크하고 true이면 첫 점과 마지막 점을 이어줌.
    - 3개 이하면, 오류 메시지를 콘솔에 띄우고 다각형을 완성하지 않음.

### 4. 화면 지우기 ( e )

- 작동 확인 방법 : c 를 눌러 확인
- 구현 방법
  - c를 눌렀을 때, 완료 여부, 회전 여부, 회전 각 등의 값을 모두 초기화.
  - PointManager 내부에 Point를 가지고 있는 벡터를 초기화.
  - 초기화 후 display 함수를 통해 화면을 띄워주어, 화면 지움.

## 5. 완성한 상태에서 SHIFT + 클릭 반응 없음 ( g )

- 작동 확인 방법 : 다각형 완성 후 SHIFT + 클릭을 통해 확인
- 구현 방법
  - isComplete 플래그를 true로 체크하고, true면 현재 위치 값을 저장하지않고 다시 화면을 보여주도록 display 함수 호출.

## 6. 방향키를 눌렀을 때, 적절한 방향만큼 이동 ( h )

- 작동 확인 방법 : 각 방향키 누르기
- 구현 방법
  - glutSpecialFunc에 등록한 special 함수에서 방향키를 눌렀을 때, 각 방향에 맞게 적절한 함수를 호출하도록 설정
    - 오른쪽을 눌렀을 때, pointManager.moveRight();
    - 왼쪽을 눌렀을 때, pointManager.moveLeft();
    - 위쪽을 눌렀을 때, pointManager.moveUp();
    - 아래쪽을 눌렀을 때, pointManager.moveDown();
  - 이렇게 설정 된 함수들은 각 방향마다 적절히 5.0 만큼 평행이동 하도록 좌표 값을 업데이트. 그 후 display를 통해 변경된 좌표값을 화면에 표시

## 7. 오른쪽 마우스 버튼 클릭 + 커서 이동 시, 움직이는 양 만큼 이동 ( i )

- 작동 확인 방법 : 오른쪽 마우스 버튼 클릭 + 커서 이동
- 구현 방법
  - glutMouseFunc에 등록한 mousepress 함수에서 오른쪽 마우스 버튼이 클릭되었을 때, 이전 좌표 값을 설정하고 rightButtonPressed라는 플래그를 true로 체크
  - glutMotionFunc에 등록한 mousemove 함수에서 오른쪽 클릭이 된채로 이동중인 지 체크하고, 현재 좌표와 이전 좌표 값의 차이만큼 pointManager.move(dx,dy)를 통해 좌표값 업데이트
  - 현재 좌표를 이전 좌표로 세팅 후 display를 통해 변경된 좌표값을 화면에 표시
  - 오른쪽 마우스가 놓아지면 ( GLUT\_UP 이면 ), rightButtonPressed를 false로 초기화.

## 8. 무게중심 점 둘레로 회전 (j, k, l)

- 작동 확인 방법 : r키를 눌러 토글
- 구현 방법
  - 무게 중심 계산

아래의 공식을 구현하여, pointManager.getCenterPoint를 통해 값을 얻어서 사용.

$$A = \frac{1}{2} \sum_{i=0}^{N-1} (x_i y_{i+1} - x_{i+1} y_i)$$

$$c_x = \frac{1}{6A} \sum_{i=0}^{N-1} (x_i + x_{i+1}) (x_i y_{i+1} - x_{i+1} y_i)$$
$$c_y = \frac{1}{6A} \sum_{i=0}^{N-1} (y_i + y_{i+1}) (x_i y_{i+1} - x_{i+1} y_i)$$

- 무게 중심을 둘레로 회전
  - glutTimerFunc(10, timer, 0) 를 통해 r키를 눌렀을 때, timer함수가 작동하도록 함. timer가 동작할 때 마다, angle 증가.
  - r키가 회전하지 않는 상태에서 처음 눌렀을 때, isRotating을 true로 세팅하고 display를 호출
    - display 함수에서 isRotating이 true이면 centerPoint를 가져와서 그 점을 노란색으로 찍어줌.

- false라면, 화면에 그리지 않게 설정하여 회전이 멈추었을 때 무게중심이 화면에서 사라짐.
- isRotating이 true라면 timer내에서 재귀적으로 glutTimerFunc를 호출
- 좌표 계산에서 무게 중심에 대한 상대적인 좌표를 구하고, 아핀 변환을 통해 좌표를 변경하여 angle 에 따라 회전하도록 설정
  - 아래의 공식을 사용
 
$$x = x\cos(\theta) - y\sin(\theta), y = x\sin(\theta) + y\cos(\theta)$$
- 회전도중 r키를 누르면 isRotating을 false로 변경하고, 재귀호출에서 탈출.

## 9. 회전시 다른 기능 불가 ( m )

- 작동 확인 방법 : r키를 눌러 회전을 시킨 후, 다른 동작 확인
- 구현 방법
  - isRotating 플래그를 체크하고, true면 다른 동작들을 하지않고 다시 화면을 보여주도록 display 함수 호출.

## 10. 윈도우 크기 변경시 윈도우의 내용 변경 ( n )

- 작동 확인 방법 : 마우스를 통해 윈도우 크기 변경
- 구현 방법
  - 아핀 변환을 이용하여 각 좌표에 윈도우 크기 변경에 대한 비율을 곱하여 좌표 계산
  - 이전 윈도우크기와 현재 윈도우 크기에 대한 비를 통해 비율을 계산하고, 비율을 통해 점의 위치 다시 계산
    - pointManager.updateRatio 함수를 통해 각 점에 비율을 곱하여 점의 위치를 다시 계산.

## 11. 프로그램 종료 ( o )

- 작동 확인 방법 : f 키 누르기
- 구현 방법

- glutKeyboardFunc에 등록한 keyboard 함수에서 f를 눌렀을 때, glutLeaveMainLoop()을 실행하도록 설정.

## 12. 프로그램 구성 요소 색 구분 ( p )

- 작동 확인 방법 : 각 요소들을 그려본다.
- 구현 방법
  - glColor3f(r,g,b) 를 통해 각 요소 ( 점, 선, 무게중심, 배경화면 ) 를 화면에 그릴 때, 적절한 색으로 설정