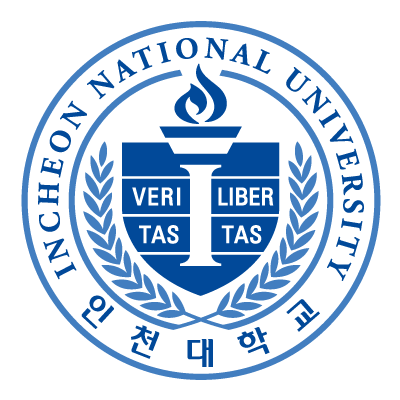
**컴퓨터 그래픽스**

**HW 1**



**컴퓨터 그래픽스**

**김지범 교수님**

**컴퓨터 공학부**

**202201529**

**정상혁**

**목차**

1. 과제 1
2. 과제 2
3. 과제 3
4. 과제 4
5. 마치며
6. 과제1

텍스트, 폰트, 화이트, 흑백이(가) 표시된 사진

자동 생성된 설명

스크린샷, 라인, 도표, 텍스트이(가) 표시된 사진

자동 생성된 설명스크린샷, 라인, 직사각형, 텍스트이(가) 표시된 사진

자동 생성된 설명

그림을 오른쪽과 같이 그려주았고, 코드로 옮기기 전에 세계좌표를 적어주었다.

그림을 그리고 코드로 옮겨 그림을 그리기 위하여 삼각형으로 나누어 주었다.

삼각형을 나누는 기준은 교수님께서 설명하셨던 대로 나오는 삼각형이 최대한 적게, 최대한 크게, 그리고 마지막으로 orientation을 일치시키며 나누어주었다.

이렇게 나누게 된다면, 삼각형은 10개가 나오게 된다.

보기 편하게 하기 위해 가시공간의 right 값은 도형의 최대 x좌표에 10 더한 값을 top 값은 도형의 최대 y좌표 값에 10을 더하여



이렇게 설정해 주었다.

그렇게 위에서 정한 값을 기준으로 orientation을 생각하며 코드로 옮겨 보았다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

그리고 orientation 방향이 BACK이기 때문에 GL\_BACK으로 옵션을 주고 실행시켰다.

삼각형은 잘 나누어 졌지만, 도형의 안이 채워지지 않았다는 걸 깨닫고, 먼저 도형을 그리고 그 위에 검은색으로 선을 긋기로 생각했다.

도표, 라인, 종이접기이(가) 표시된 사진

자동 생성된 설명

코드를 실행 한 후 나오는 출력화면이다.

1. 과제 2

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

스크린샷, 텍스트, 디자인이(가) 표시된 사진

자동 생성된 설명1) 왼쪽처럼 코드를 바꾸고 출력 결과는

이렇게 나오게 된다. 그 이유는

GL\_POLYGON의 랜더링 방식에 있다.

GL\_POLYGON의 랜더링 방식에 따라

orientation을 맞추어서,

첫번째 vertex를 기준으로,

첫번째 삼각형은 v0, v1, v2로

스크린샷, 라인, 그래프, 폰트이(가) 표시된 사진

자동 생성된 설명두번째 삼각형은 v0, v2, v3로

세번째 삼각형은 v0, v3, v4로 그리게 되고 이 경우는 v4의 색깔을 빨간색으로 줬으므로 3와 v0까지의 색깔이 보간을 이용해 그라데이션으로 색깔이 나오게 된다.

텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명2) b의 출력결과는 오른쪽과 같이 나오게 된다.

오른쪽과 같이 나오는 이유는 아래 그림과 같은데, 이것도 마찬가지로 GL\_POLYGON으로 렌더링을 하므로 첫번째 vertex인 v0를 기준으로

첫번쨰 삼각형은 v0, v1, v2로,

두번째 삼각형은 v0, v2, v3로,

스크린샷, 라인, 도표, 그래프이(가) 표시된 사진

자동 생성된 설명세번쨰 삼각형은 v0, v3, v4로 그려지게 된다.

색깔의 경우에는 v3가 빨간색으로 생기게 되고, 이것 역시 보간을 이용해 색을 칠하게 된다.

3) a와 b의 차이점은 색의 차이점을 볼 수 있는데, 이 차이가 일어나는 이유는,

a와 b 둘 다 빨간색인 점이 왼쪽 위의 꼭짓점이라는 점은 같지만,

a의 경우는 (80, 80), (20,20) 과 삼각형이기 때문에 삼각형 하나가 그라데이션이 적용된 반면,

b의 경우에는 (20, 20), (50, 20)과 이루는 삼각형과 (50, 20), (80, 80)과 이루는 삼각형 둘 다 있기 때문에 색 보간 역시 이 두삼각형에 맞춰서 이루어지게 된다.

따라서 a와 b의 색깔의 차이가 나타나게 된다. 그리고 a가 b에 비해 long Thin 삼각형이 많아 보이기 때문에 b가 더 잘 나누어졌다고 볼 수 있다.

1. 과제 3

텍스트이(가) 표시된 사진

자동 생성된 설명

1) 더 좋은 건 (a) triangulation 이라고 생각한다 그 이유는 (b)에 비해 long thin 삼각형들이 더 적고 또한 삼각형들의 크기가 더 크고 일정하기 때문이다.

2) 이 도형은 정12각형이다. 이 꼭짓점들의 세계 좌표를 정하기 위해, 각도가 30도씩 늘어난다는 것을 이용하여, sin과 cos을 이용하여 계산해주었다. 다음은 중점이 (0, 0)이라고 가정하고 구한 것이다.

텍스트, 스크린샷, 도표, 원이(가) 표시된 사진

자동 생성된 설명

하지만 내가 설정한 가시범위는 (0, 100, 0, 100, -1, 1) 이므로 조정이 필요하다.

고로 중점을 (50, 50) 으로 옮기고 계산식을 조정하여 for문을 사용해 코드를 작성해준다면,

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

위와 같은 코드가 나오게 된다. 이 코드는 b와같이 정십이각형의 오른쪽 꼭짓점을 기준으로 랜더링이 되게 된다.

또한 조금 더 직관적으로 확인하기 위해 삼각형들로 나누어 주는 코드를 추가하고 코드를 실행해보면 아래와 같은 출력 결과가 나오게 된다.

도표, 종이접기이(가) 표시된 사진

자동 생성된 설명

1. 과제 4

텍스트, 스크린샷, 폰트, 직사각형이(가) 표시된 사진

자동 생성된 설명

스크린샷, 라인, 도표이(가) 표시된 사진

자동 생성된 설명

나는 steiner vertex로 (40, 10)과 (40, 50)으로 설정하여 삼각형으로 나누어 주었다.

그 이유는 저 위치가 가장 long thin 삼각형이 나오지 않는 위치라고 판단했기 때문이다.

도표, 라인, 직사각형, 디자인이(가) 표시된 사진

자동 생성된 설명Orientation이 Back 방향이므로 glPolygonMode를 BACK으로 주고 렌더링을 해주었다.

그러자 내가 생각한대로

랜더링이 잘 된 것을

확인하였다.

1. 마치며

이번 과제를 통해 opengl이라는 도구를 좀 더 잘 다룰 수 있게 된 것 같고, 또한 지금까지

배웠던 것들을 종합적으로 복습하는 계기가 된 것 같다. 수업을 들을 때는 마냥 그렇구나

하고 넘어갔던 것들을 막상 과제로 해보니 내가 진짜로 이해한 것이 아님을 깨닫고

내 부족함을 많이 느낌과 동시에 성장을 할 수 있었던 것 같다.

과제를 하기 위해 에러도 나고, 이에 대한 해결책을 찾기 위해 주도적으로 고민도 해보며

원하는 결과값이 나올 때 뿌듯하면서 성취감이 같이 들었다. 이런 과정을 통해서

더욱 실력이 늘 수 있게 된 것 같아서 컴퓨터 그래픽스에 대한 흥미를 더욱 느낄 수 있게 되었다.