

컴퓨터 그래픽스

Draw Program

소프트웨어학과 32170533

김민준

1. 개요

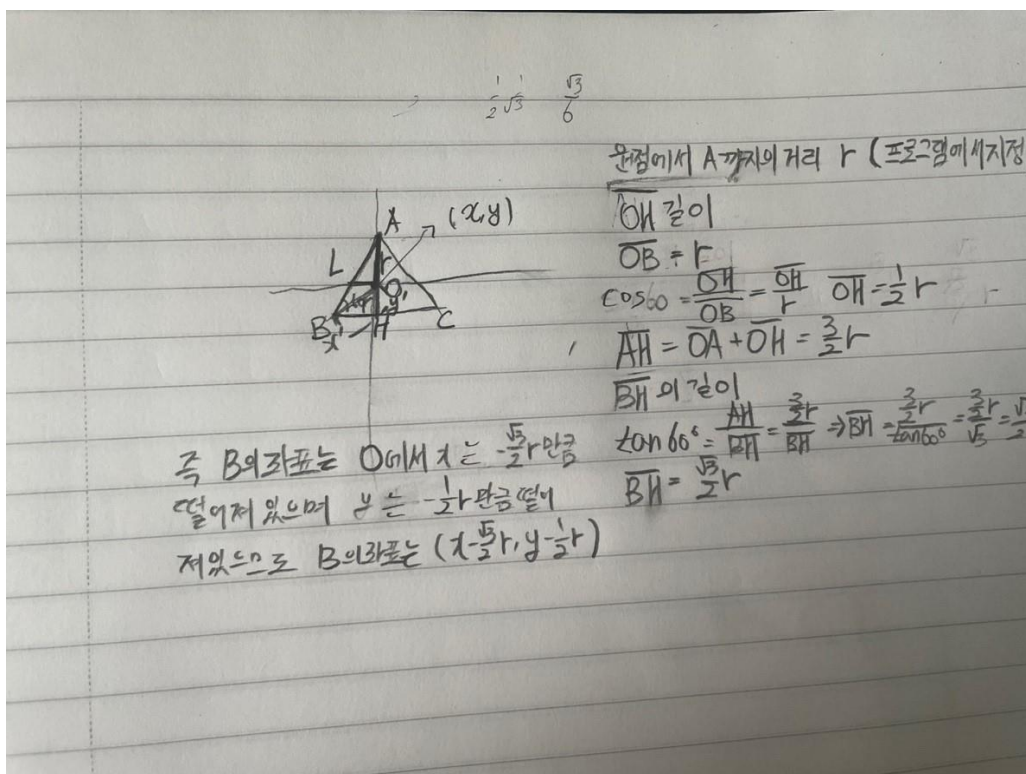
이번 과제는 무언가를 그리는 프로그램을 만드는 것입니다. 저는 이번 과제에 삼 각형을 그리는 프로그램을 만들었습니다. 삼각형을 그릴 때, 삼각형의 크기, 색깔, 회전하는 것이 가능합니다.

2. 기능

캔버스 안을 클릭했을 때, 커서 주변으로 일정한 부분에 삼각형이 생성됩니다.\

삼각형 그리기

- 정삼각형의 경우 각 변의 길이가 같은 삼각형을 의미합니다. 그러므로 제 1번 클릭한 커서의 위치로부터 각 꼭짓점의 길이는 같습니다. 만약 원점이 (x, y) 고 각 꼭짓점의 좌표를 구하면 다음 그림과 같습니다.



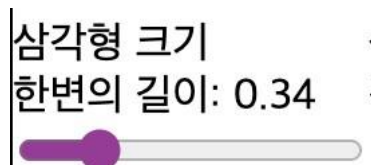
- B의 좌표의 x 좌표는 $(x - \frac{\sqrt{3}}{2}r)$ 이며, y 좌표는 $(x - \frac{1}{2}r)$ 입니다. 위 계산을 토대로 c의 좌표를 구하면 결국 HC 선분의 길이도 $\frac{\sqrt{3}}{2}r$ 이므로 c의 좌표의 x 좌표는 $(x + \frac{\sqrt{3}}{2}r)$ 이며, y 좌표는 $(x - \frac{1}{2}r)$ 또한 A의 좌표는 x 좌표는 원점과 같고, y 좌표는 원점의 y 좌표에서 r 만큼 더한 값이 됩니다.
- r의 경우 제가 사용자가 입력 값을 지정할 수 있습니다. 예를 들어 r을 100으로 설정한다면. 클립좌표계의 값에 따라 100으로 나눈 값이 r이 됩니다. 또한 이 r이 설정 되었을때 삼각형 한변의 길이는 $\sqrt{3}r$ 이 됩니다.

```
let leftBottomX = centerX - triangleSize * (Math.sqrt(3) / 2)
let leftBottomY = centerY - triangleSize * (1/2)
let rightBottomX = centerX + triangleSize * (Math.sqrt(3) / 2)
let rightBottomY = centerY - triangleSize * (1/2)
let middleX = centerX
let middleY = centerY + triangleSize
```

- (centerX, centerY)가 원점이고 triangleSize가 r이 됩니다.

삼각형의 크기

- 이 프로그램에서는 삼각형의 크기도 지정할 수 있습니다. 삼각형의 크기는 위에서 말했던 r을 의미합니다.



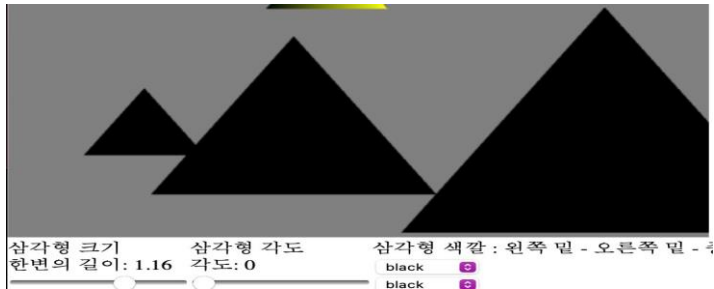
- 위 그림에서 input 태그의 type=range를 이용하여 값을 받습니다. 위 슬라이더에 따라 삼각형의 한변의 길이를 지정할 수 있습니다. 이 값을 shader에 전달하는 방법은 이벤트 리스너를 사용하는 것입니다. onChange 이벤트를 이용하여 전달하게 되는데.

triangleLineLength => 전역변수 (r * 100을 의미)

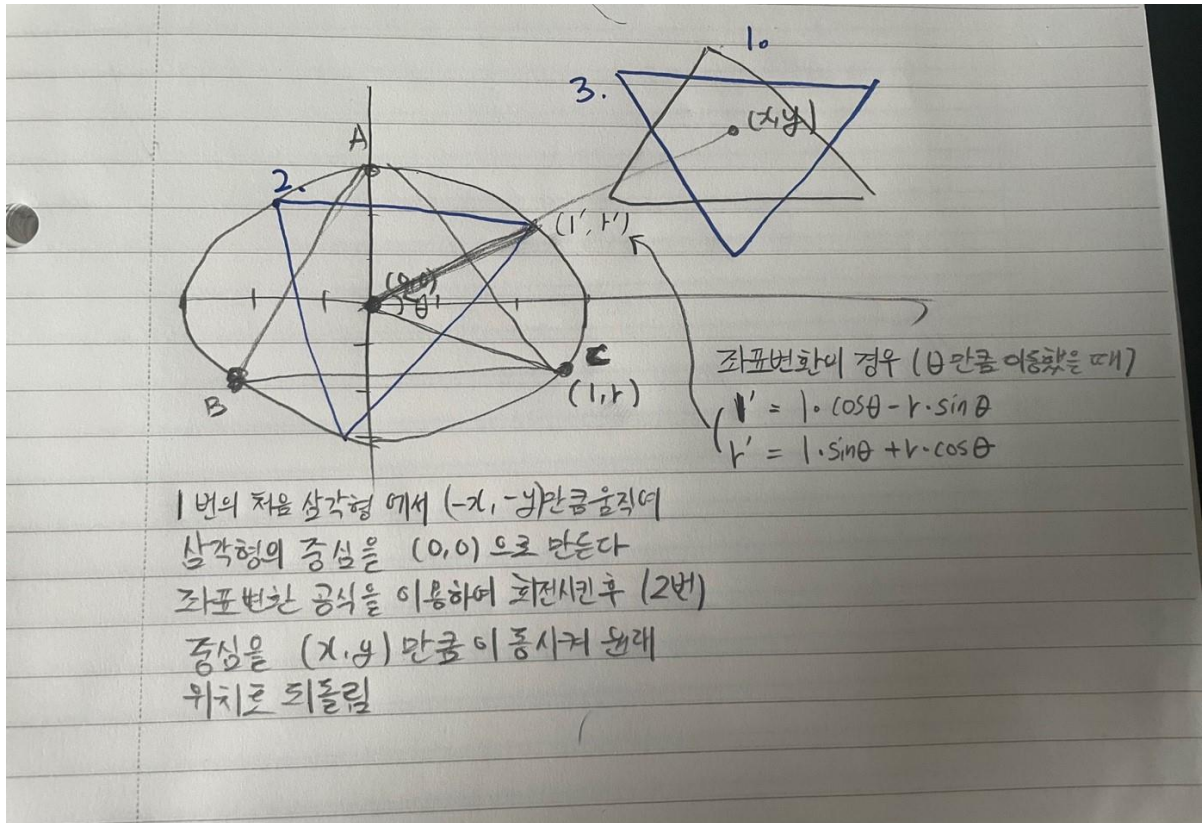
```
<input type="range" id="slide" min="0" max="100" value="20">

slide.addEventListener("change", (e) => {
  triangleLineLength = e.target.value
  rangeValue.innerText = (e.target.value / 100 * Math.sqrt(3)).toFixed(2)
})
```

- 다음과 같이 id 가 slide 인 노드를 가져와 슬라이더의 값이 변할 때 마다 전 역변수 triangleLength 를 바꿔줍니다. 이 값을 토대로 삼각형의 크기를 지정할 수 있습니다.



삼각형의 회전



- 삼각형의 회전의 경우 θ 의 값에 따라 원점을 기준으로 회전하는 것을 말한 다. 우리가 삼각형을 회전시키기 위해서는 한점만 회전 시키는 것이 아니라 삼각형 3 점을 모두 회전시켜야한다. 또한, 삼각형을 회전시키기 위해서 는 클릭한 좌표에서 회전시키는 것이 아니라 원점으로 우선 이동해서 회 전시킨 후, 원래의 클릭 좌표로 이동한다.

```
function rotatePoint(center, rotatePoint, deg) {

    let x = rotatePoint[0] - center[0]
    let y = rotatePoint[1] - center[1]

    let rotateX = x * Math.cos(deg) - y * Math.sin(deg)
    let rotateY = x * Math.sin(deg) + y * Math.cos(deg)

    return [rotateX + center[0], rotateY + center[1]]
}
```

- rotatePoint 에서 원점의 center(삼각형의 중심)만큼 빼주어 원점에대한 삼각 형 A, B, C 좌표를 생성해준다. 그 다음 회전 공식을 이용하여 좌표를 회전 시킨 후 원래 클릭한 좌표로 다시 이동시킨다.



- 위 그림과 같이 삼각형을 회전시킬 수 있습니다.

삼각형의 색 설정

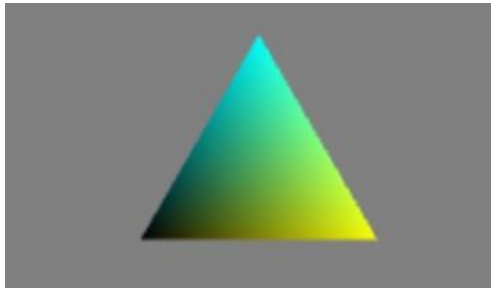
- 삼각형의 경우 정점마다 색을 가지고 있다. 그러므로 각 정점에 대하여 색 을 설정할 수 있다.

```
var colors = [
    vec4( 0.0, 0.0, 0.0, 1.0 ), // black
    vec4( 1.0, 0.0, 0.0, 1.0 ), // red
    vec4( 1.0, 1.0, 0.0, 1.0 ), // yellow
    vec4( 0.0, 1.0, 0.0, 1.0 ), // green
    vec4( 0.0, 0.0, 1.0, 1.0 ), // blue
    vec4( 1.0, 0.0, 1.0, 1.0 ), // magenta
    vec4( 0.0, 1.0, 1.0, 1.0 ) // cyan
];
```

- 다음 그림에 있는 색을 정점에 적용할 수 있다.

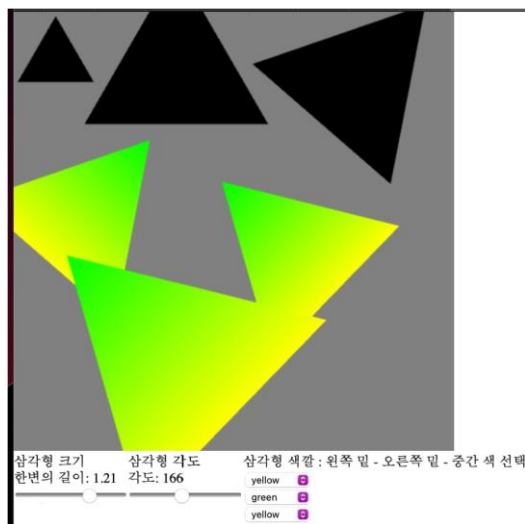
```
triangle = [
    vec4(colors[colorIndex1]),
    vec4(colors[colorIndex2]),
    vec4(colors[colorIndex3])
]
```

- colorIndex 는 내가 정점에 설정한 색이다. 이 색을 이용하여 색을 설정할 수 있는데 만약 정점에 black, yellow, cyan 을 설정하면

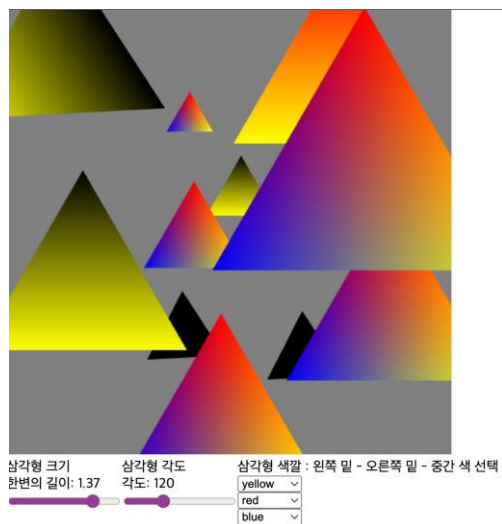


- 다음과 같은 색이 생성된다. 이런 방식으로 삼각형의 색을 커스텀하여 지정할 수 있다.

실행 결과



사파리



구글 크롬