

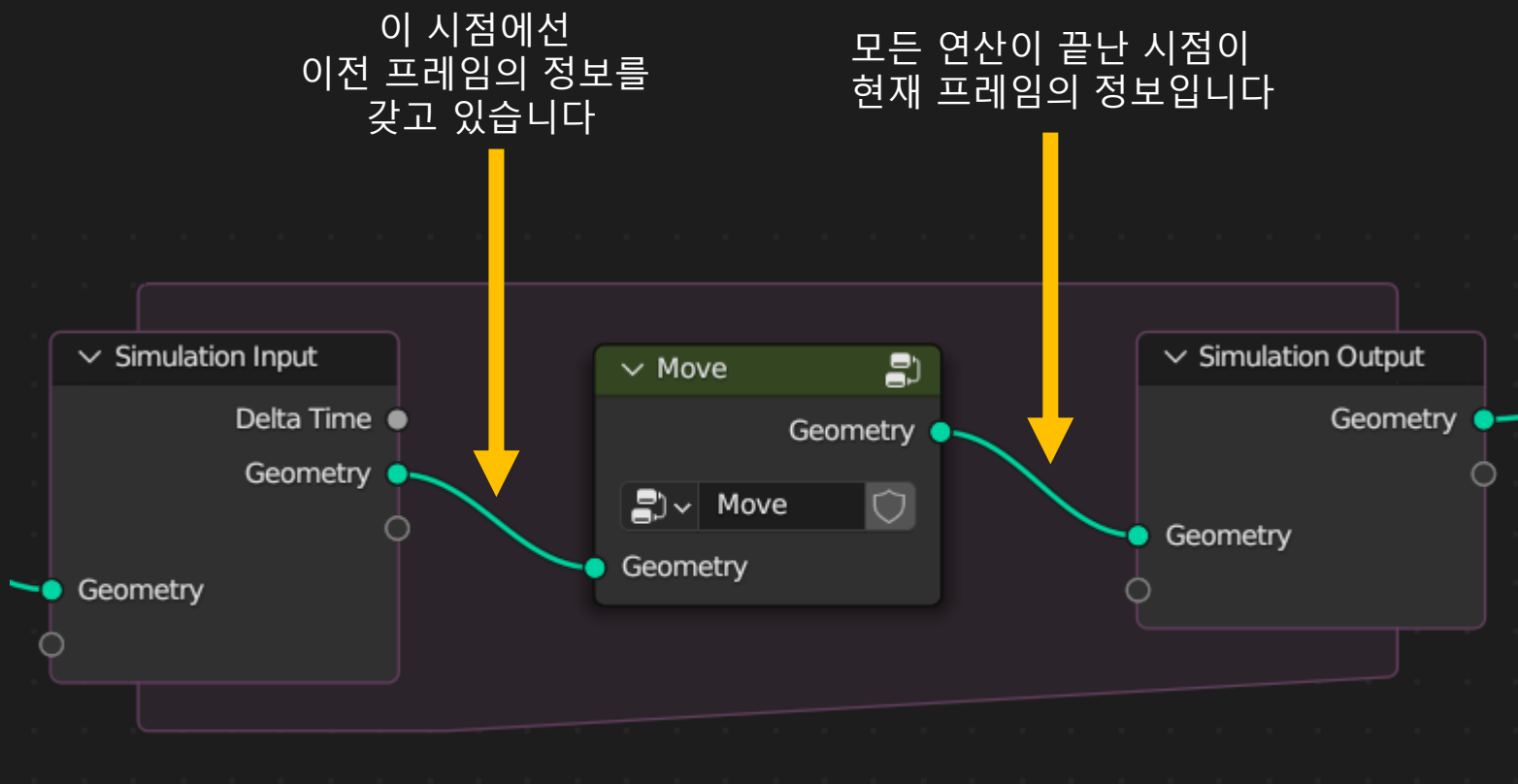
## 067강 Simulation Node – 잔상 (v3.6~)

이전 프레임의 정보를 가져오는 법



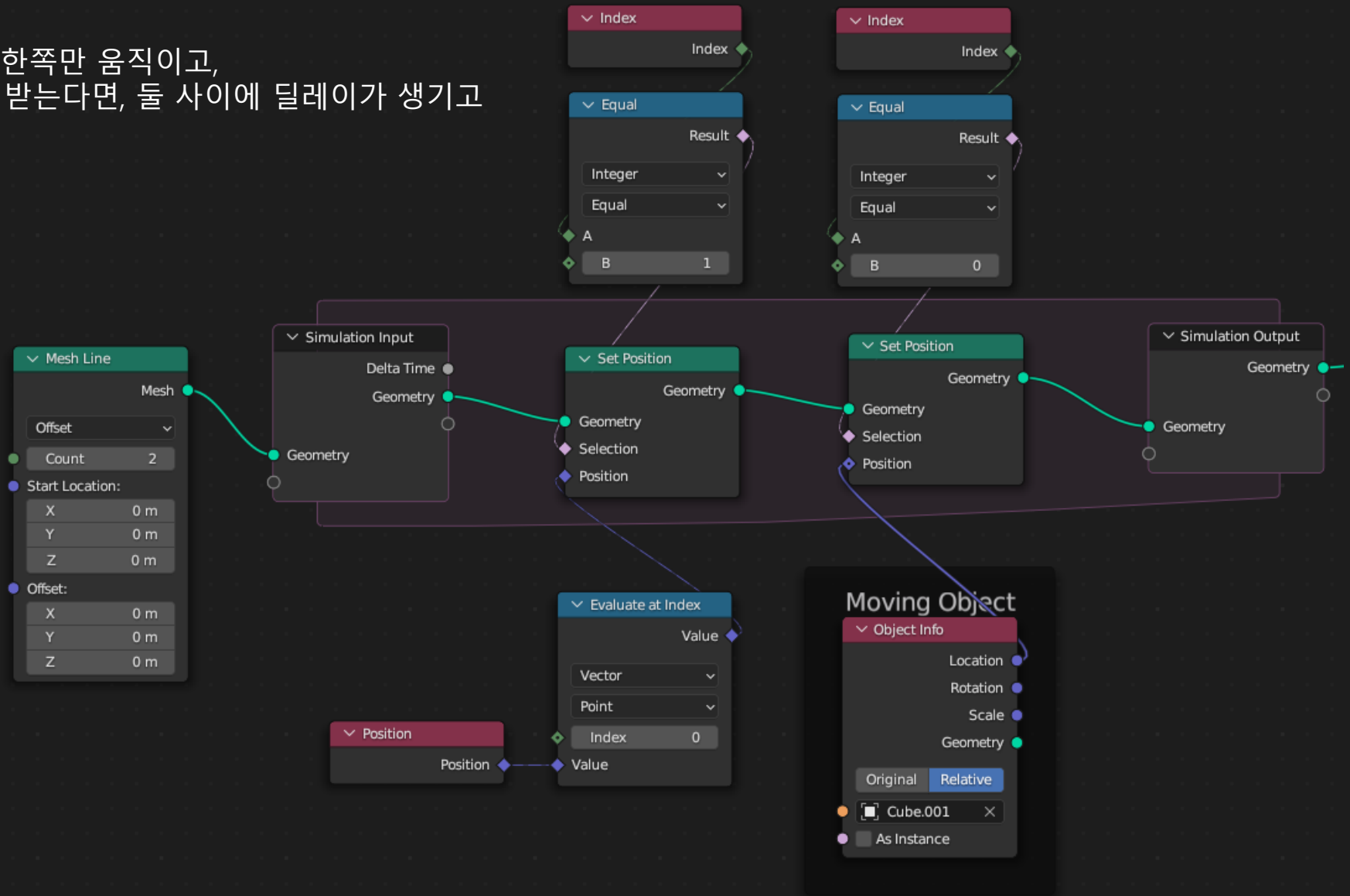
# 이전 프레임의 정보

블렌더에서는 이전 프레임의 정보를 현재 프레임에서 얻는 것이 힘듭니다.  
지오메트리 노드도 마찬가지지만, 시뮬레이션 노드는 다릅니다.



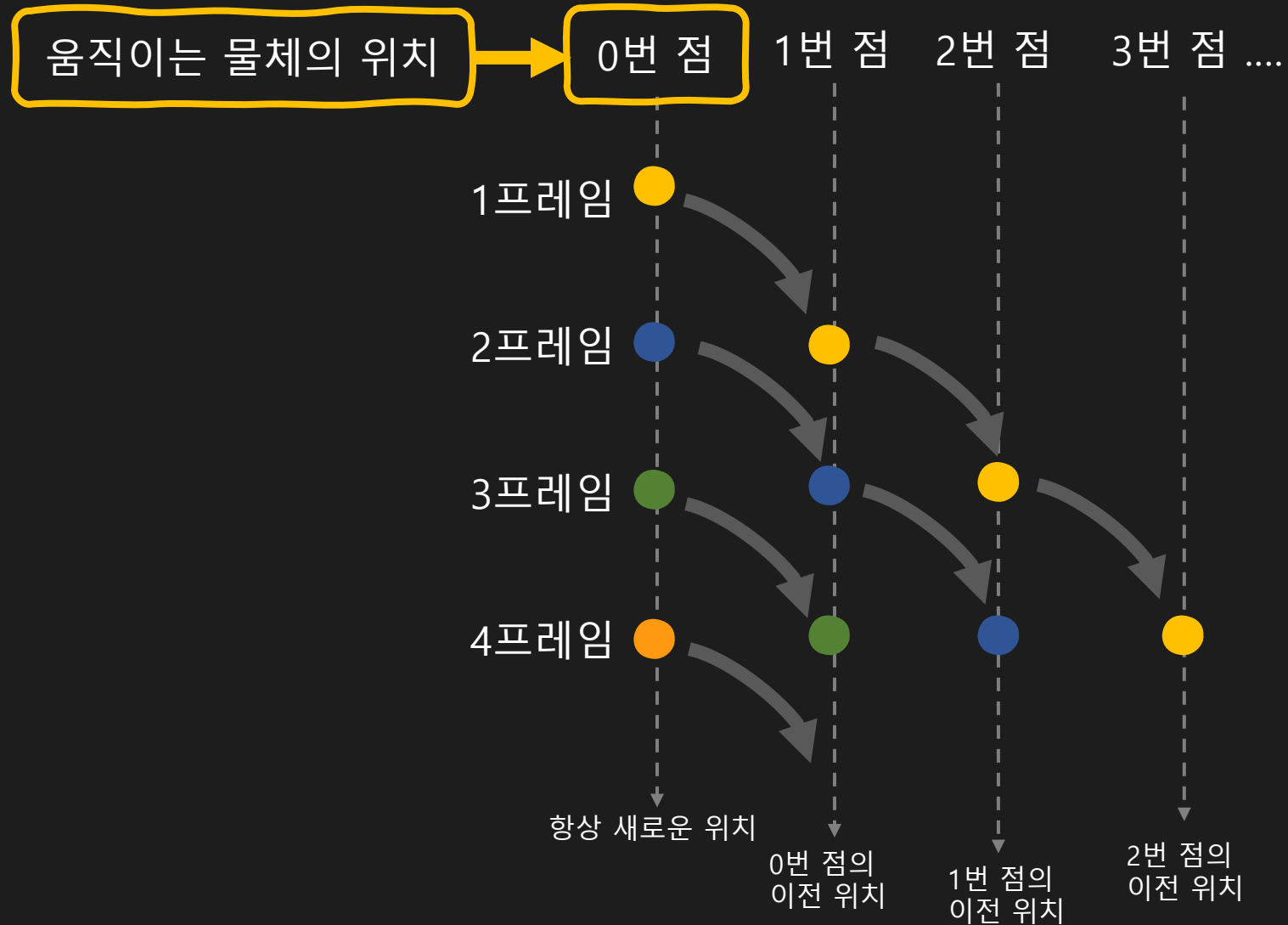
# 잔상(1)

움직이는 오브젝트의 위치를 받아, 한쪽만 움직이고,  
다른 한쪽은 움직이기 전의 위치를 받는다면, 둘 사이에 딜레이가 생기고  
잔상이 됩니다.



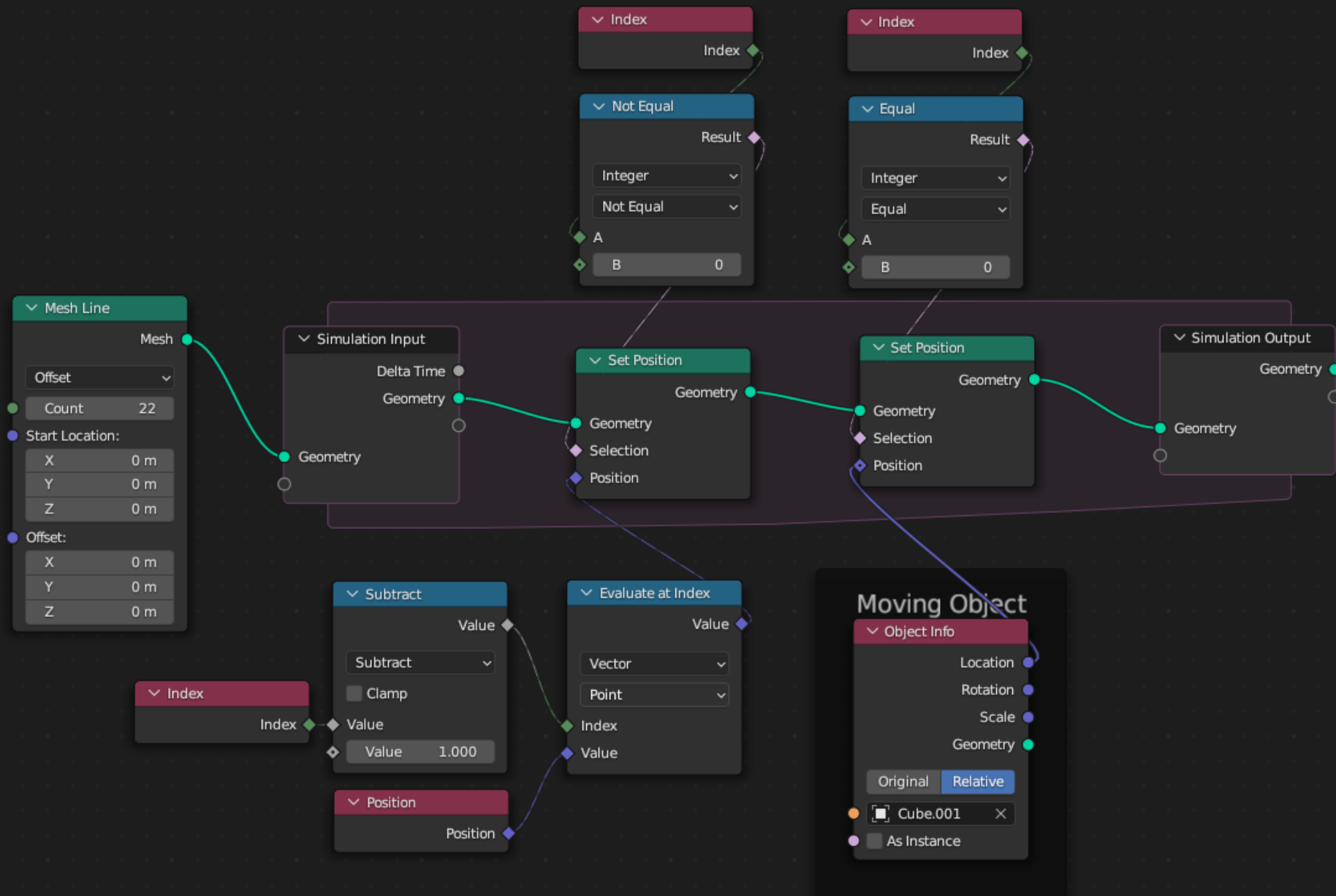
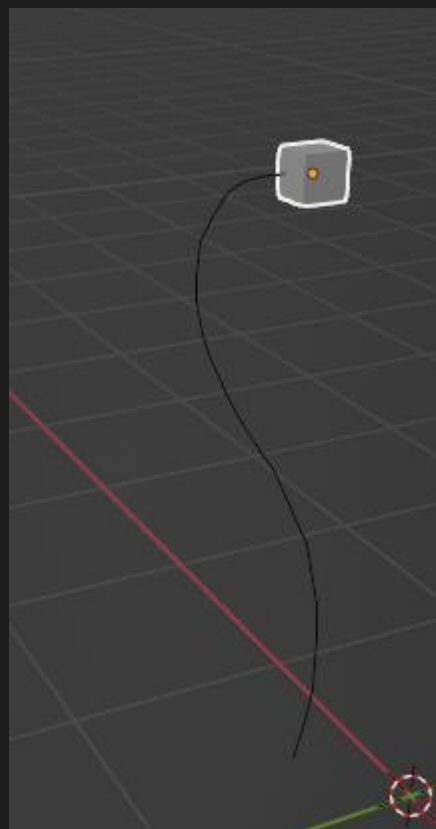
## 잔상(2)

같은 원리로, 맨 처음 포인트에만 정보를 전달한 뒤,  
나머지는 자신보다 **한칸 앞번호의 위치**를 받아오면,  
길게 꼬리를 물고 움직이게 됩니다.



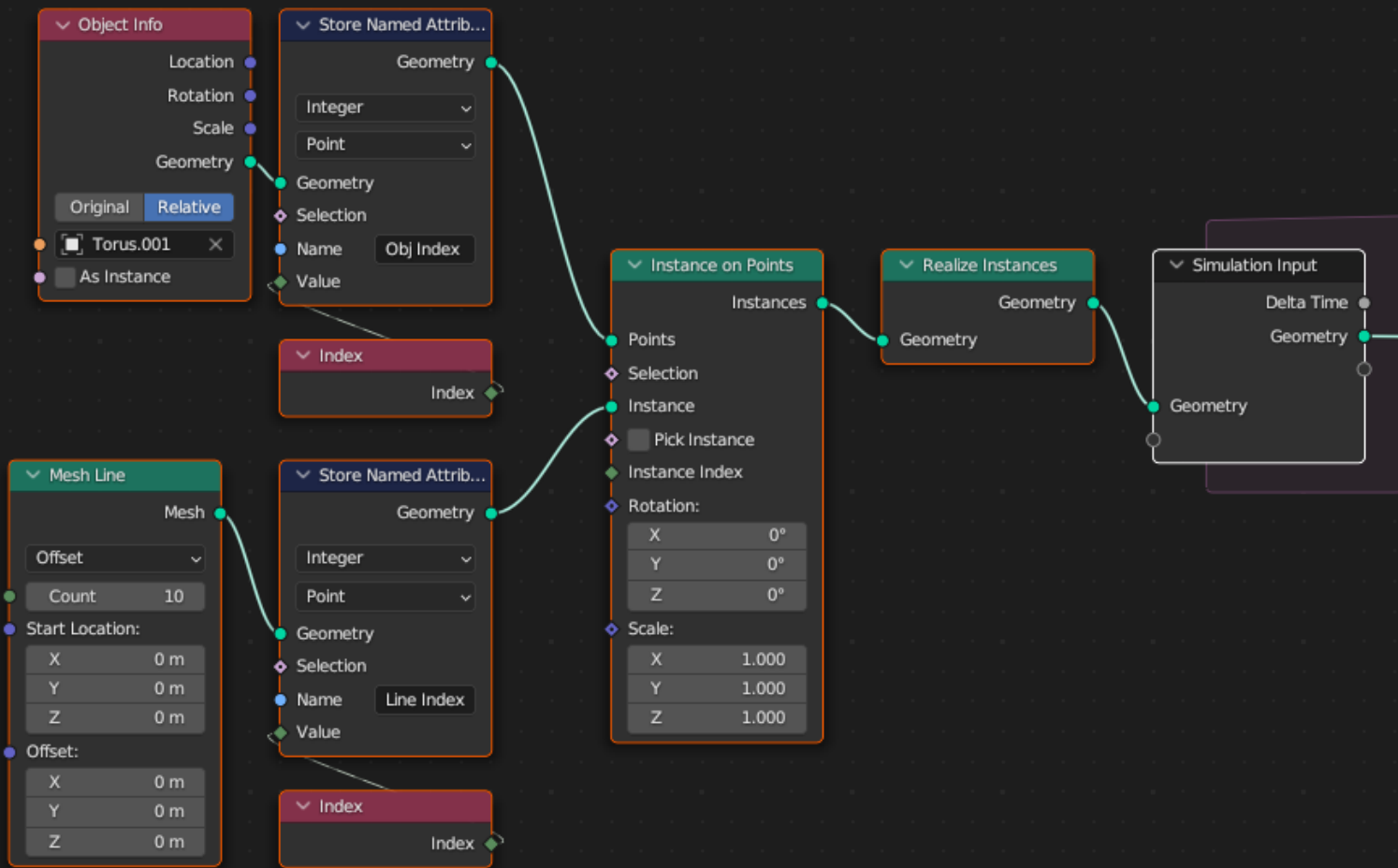
# 잔상(3)

실제 연결은 다음과 같습니다.



# 여러 점의 잔상

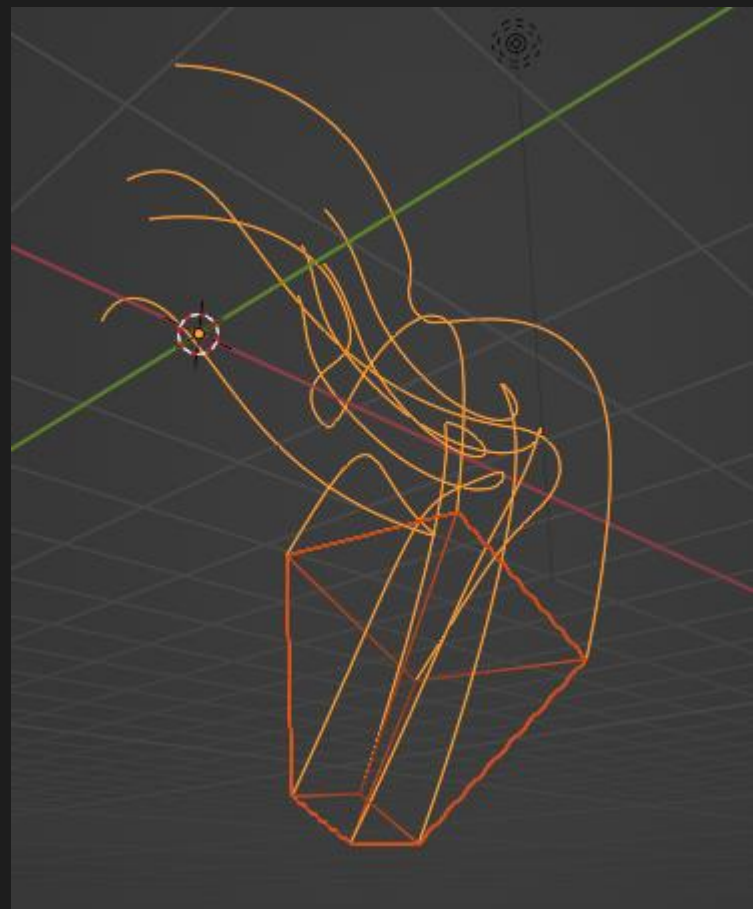
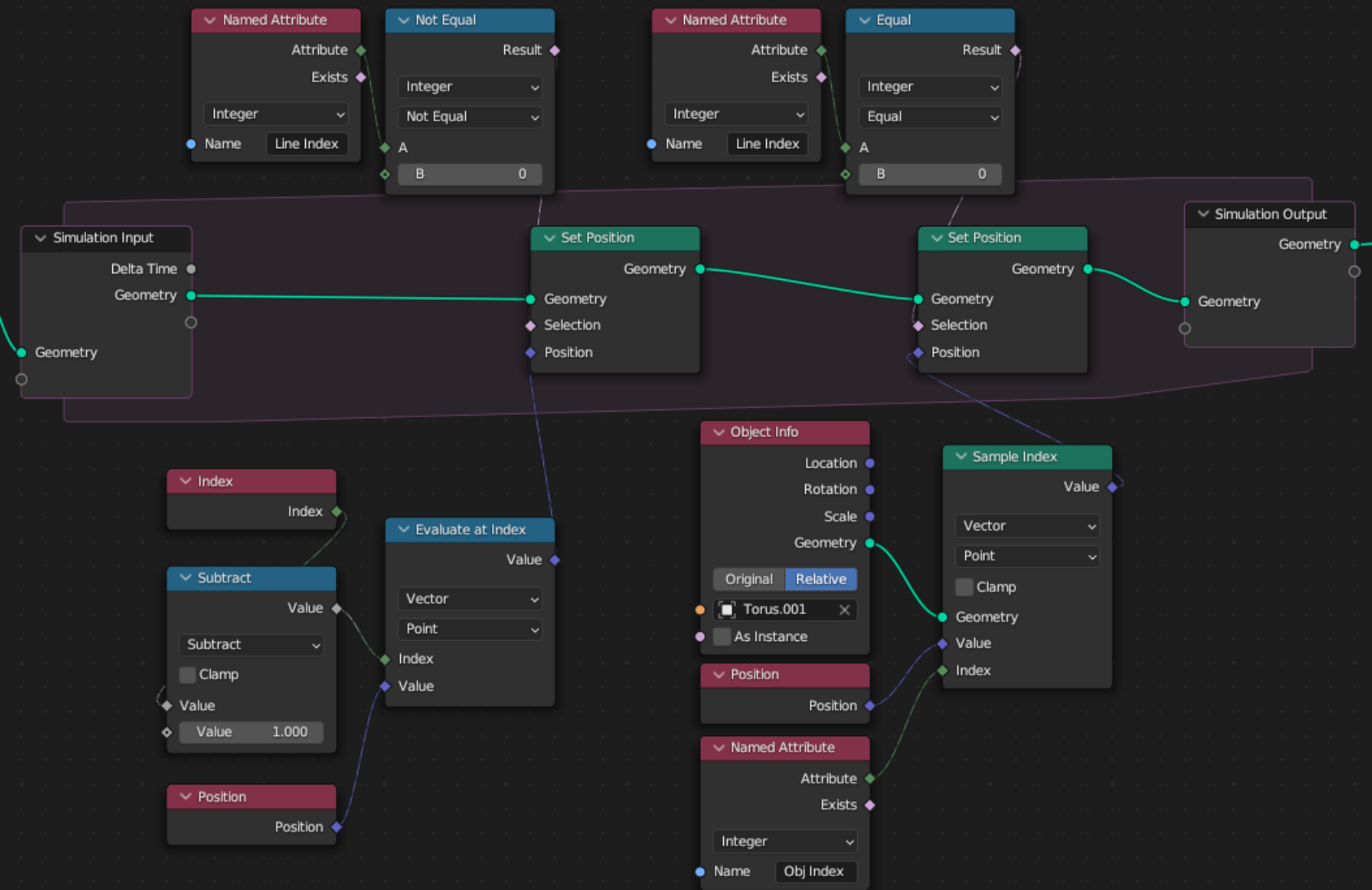
여러 개의 위치를 추적하고 싶으면 선을 여러 개 만들어야 합니다.



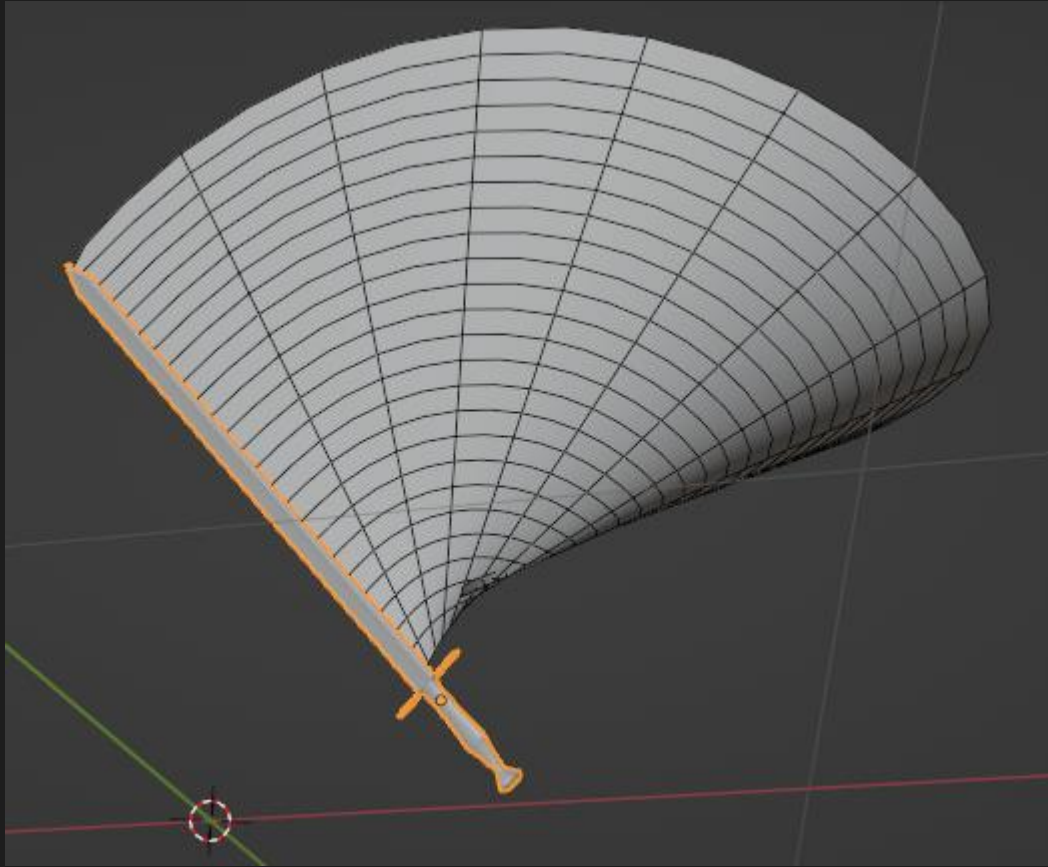
각각의 위치에 선을 심고, 인덱스를 저장합니다.  
이 인덱스는 Realize 뒤에도 유지됩니다.

# 여러 점의 잔상(2)

원래 선의 인덱스를 바탕으로 움직임을 나눠 계산합니다.



# 선의 잔상(1)



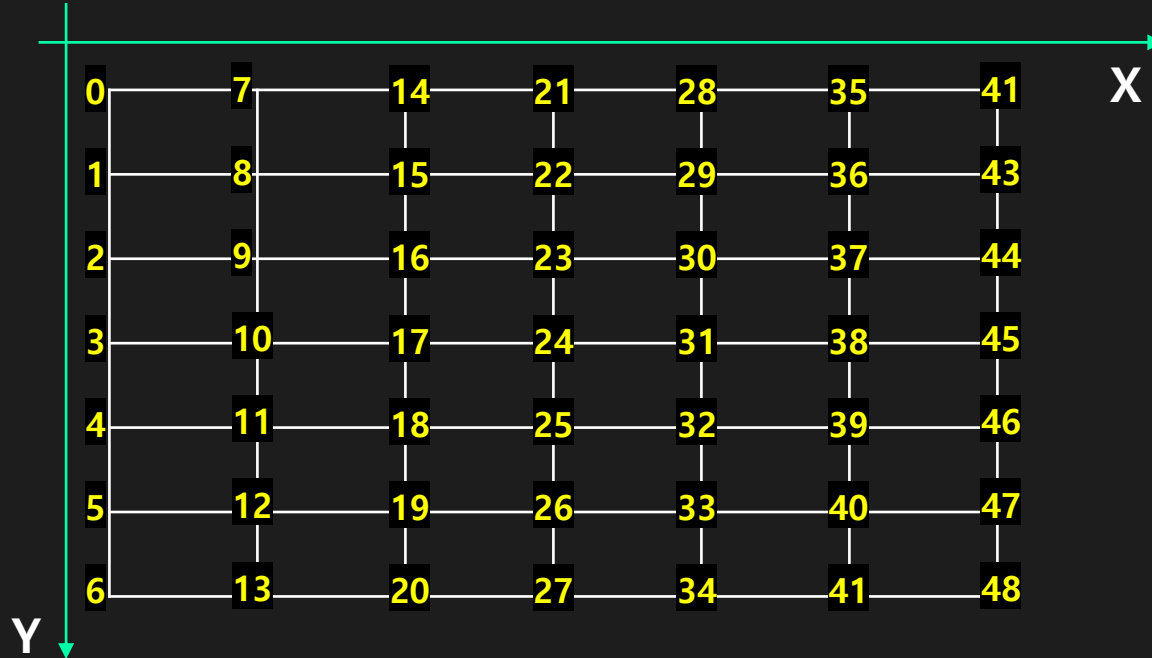
점의 잔상은 선이므로, 선의 잔상은 면이 됩니다.

기본 원리는 같지만, 2차원이 되면 인덱스를 대응시키는 것이 조금 어려워집니다.



## 선의 잔상(2)

선을 그리드에 대응시킵니다.



그리드의 7보다 번호가 작은 점들( 0~6번)은 동일 번호의 위치를 그대로 받습니다.

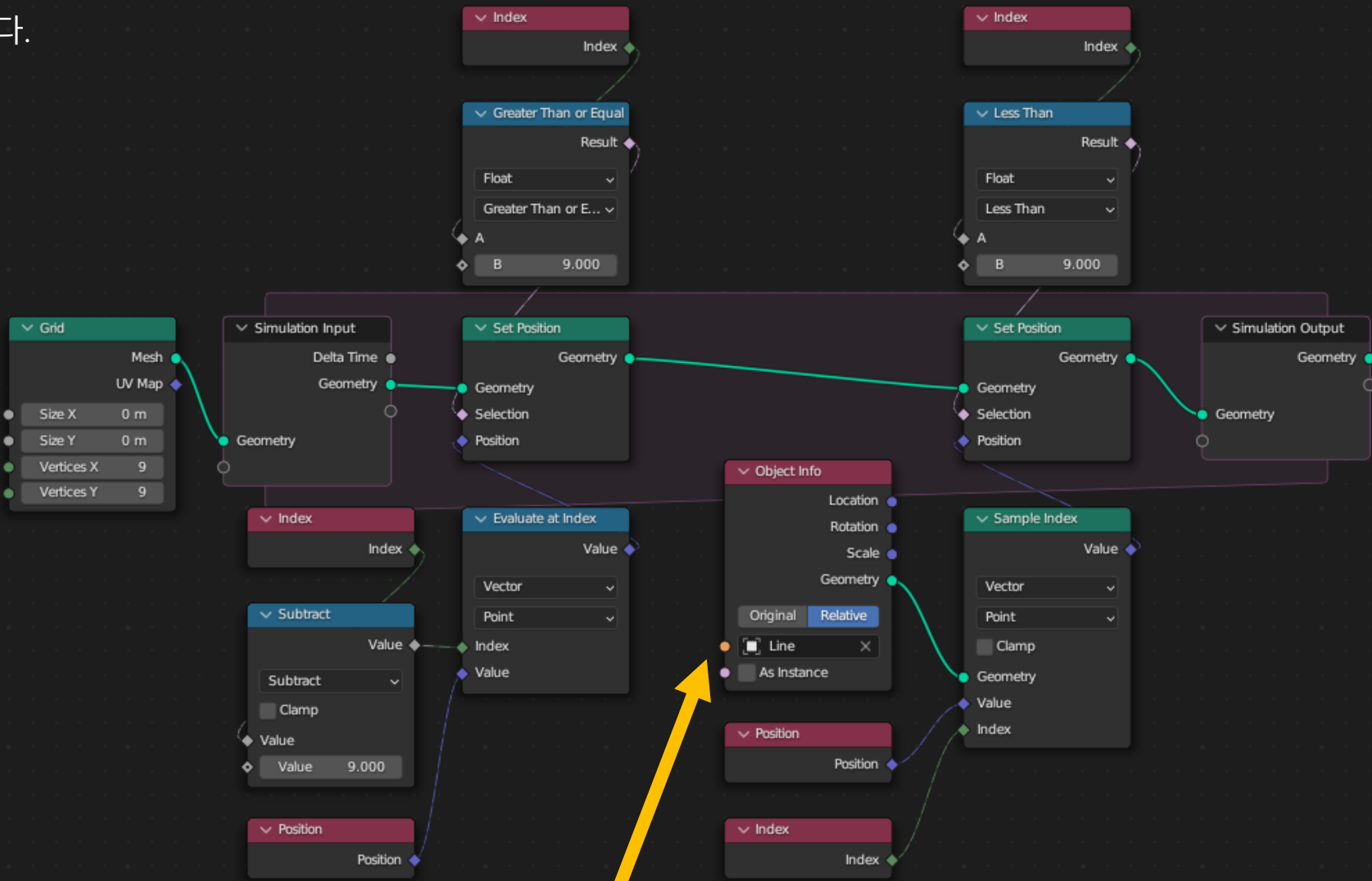
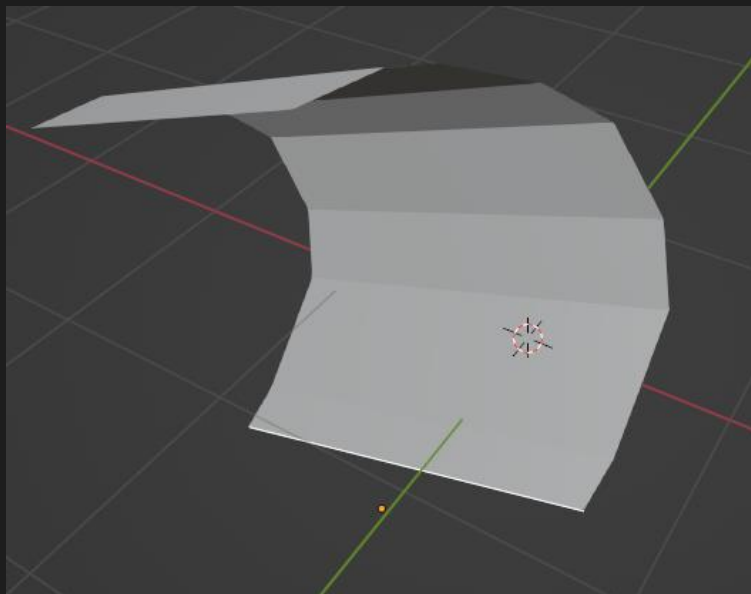
나머지 점들은 자기보다 7 작은 번호의 위치를 이어받습니다.

1에 대응되는 점 : 1,8,15,22,29,... : 7로 나누어 나머지가 1

딜레이가 2 프레임인 점 : 14,15,16,17,...20 : 7로 나누어 몫이 2

## 선의 잔상(3)

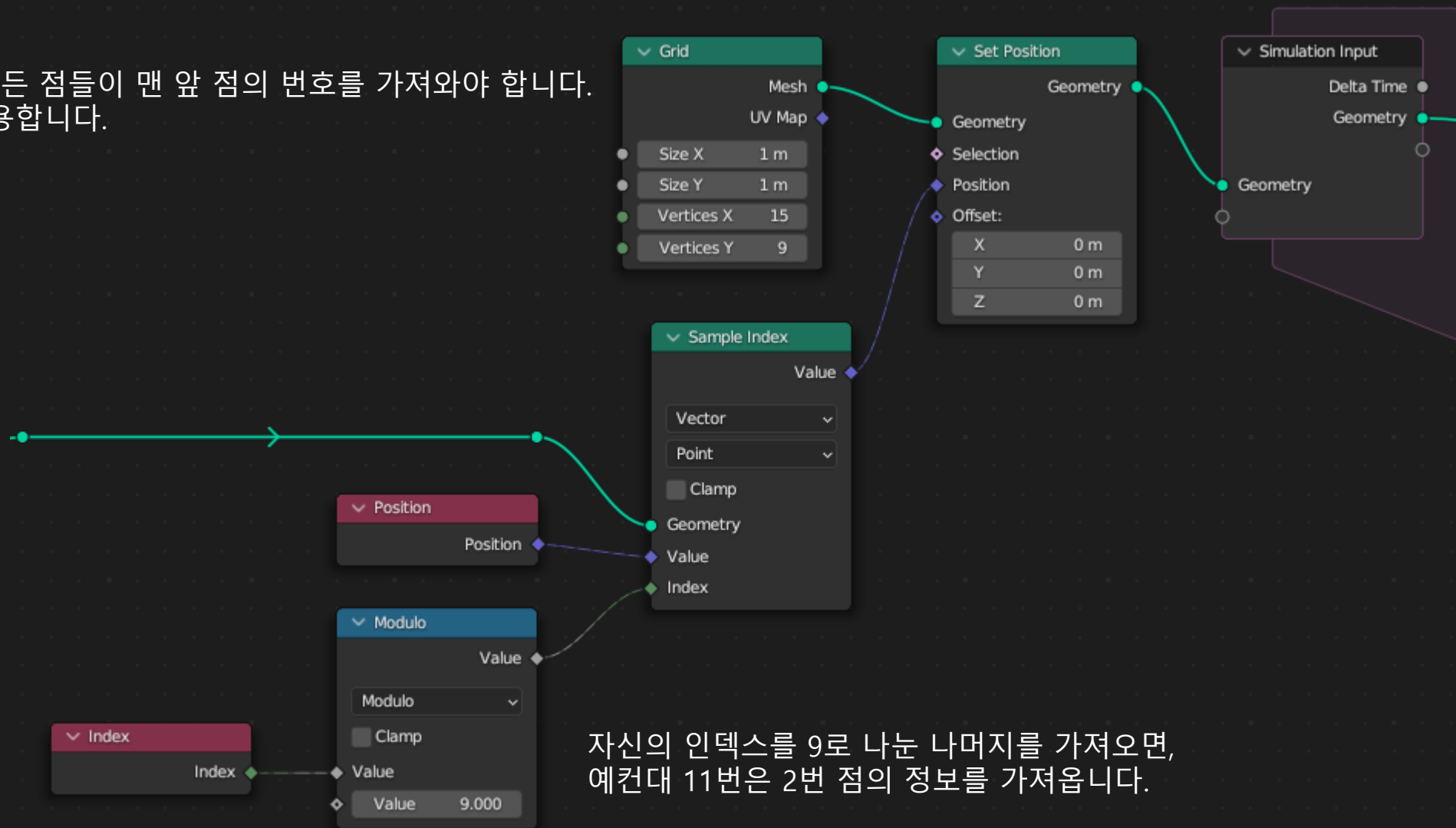
다음은 점 개수가 9개일 때의 잔상입니다.



Line은 인덱스가 순서대로여야 합니다.

## 선의 잔상(4)

초기 위치를 설정하려면 모든 점들이 맨 앞 점의 번호를 가져와야 합니다.  
이 때는 modulo연산이 유용합니다.



자신의 인덱스를 9로 나눈 나머지를 가져오면,  
예컨대 11번은 2번 점의 정보를 가져옵니다.