벡터의 내적과 평면 방정식을 이용하여 3차원 상에서의 기울기를 파악하고 이를 알리는 프로그램을 구현할 수 있을까?

3학년 2반 16번 이름 : 정휘준

Ⅰ. 연구의 필요성 및 목적

연구자 본인은 관심사 중 하나인 게임 분야 중 3D 게임에 대해 조사하다가 수학 수업 시간에 배운 벡터와 3차원 상에서의 기울기라는 내용을 접하게 되었다. 벡터는 크기와 방향을 갖는 물리량을 의미하는데, '게임 프로그래머는 수학을 배워야 할까?'라는 기사에 따르면 이는 공을 튕기는 단순한 게임부터 FPS게임, 물리 법칙이 적용된 3D 게임까지 실시간 움직임과 관련된 게임이라면 대부분 사용되는 개념이다. 그리고 '당근으로 로봇 길들이기… 구글, 로봇 보행을 위한 강화학습 기술'이라는 기사에 따르면 기울기 파악은 로봇 공학 분야에서 중요한 기술 중 하나인데, 보행을 하는 로봇들에게 필수적이며 완전하게 구현하기 어려운 기술이라고 한다. 때문에 본인은 수학 수업 시간에 배운 벡터라는 개념에 대해 '베일이 들려주는 벡터 이야기'라는 책을 통해 더 깊게 알아봤고, 이를 이용해 로봇의 기울기를 파악해 보행 로봇을 개선하는 것 뿐만 아니라, 건물, 구조물 등의 기울기를 파악해 사고를 예방하는 분야에도 기울기를 파악하는 프로그램을 적용할 수 있을 것이라는 생각이 들었고, 그것을 프로그래밍 사이트이자 프로그램인 엔트리를 이용해 구현할 수 있을 것이라는 생각이 들어 이 주제를 선정하게 됐다.

Ⅱ. 이론적 배경

1. 보행 로봇

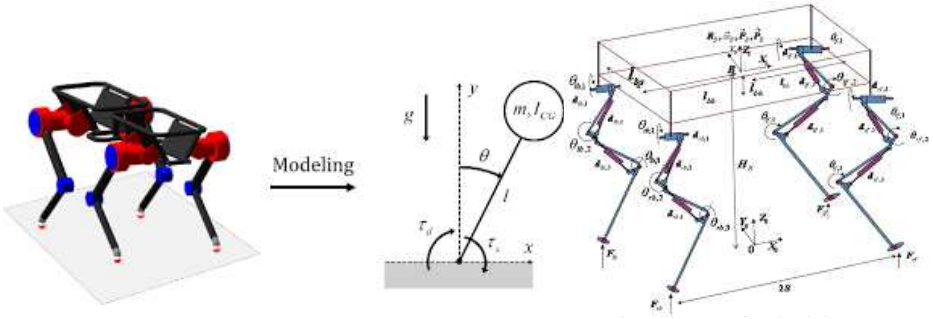
가. 휴머노이드

휴머노이드는 인간의 신체와 외형이 비슷한 로봇을 뜻한다. 즉, 두 발로 걸어 다니는 로봇이며, 최초의 휴머노이드인 1973년에 개발된 와봇-1과 1984년에 개발된 와봇-2는 걸을 수는 있었지만 매우 부자연스러웠다고 한다. 휴머노이드 로봇은 시간이 지나면서 부드러우면서 빠르게 걸을 수 있게 되었고 현재는 군용으로 4족 보행, 6족 보행 로봇까지도 개발되었다.

나. 4족 보행 로봇

4족 보행 로봇은 복잡한 지형에서의 높은 이동 성능을 핵심 기술로 주목받으면서 다양한 연구가 수행되어 왔다. 다양한 연구가 수행되었음에도, 여전히 비정형적인 지형에서의 안정적인 보행은 어려운 문제로 남아있다. 외부 충격, 지면과의 미끄러짐, 구동기의 출력 한계는 족형 로봇의 보행을 더욱 어렵게 만드는 요소들이다. 이러한 문제들을 해결하기 위해 보행 알고리즘에는 보행 패턴 생성 이외에도 자세 안정화 전략이 포함되어야 한다. 아래의 그림에서는 로봇의 다리 여러 부위에 각도를 조절할 수 있는 관절이 장치되어 있다는 것을

알 수 있다. 즉, 이 연구의 결과로 산출될 프로그램

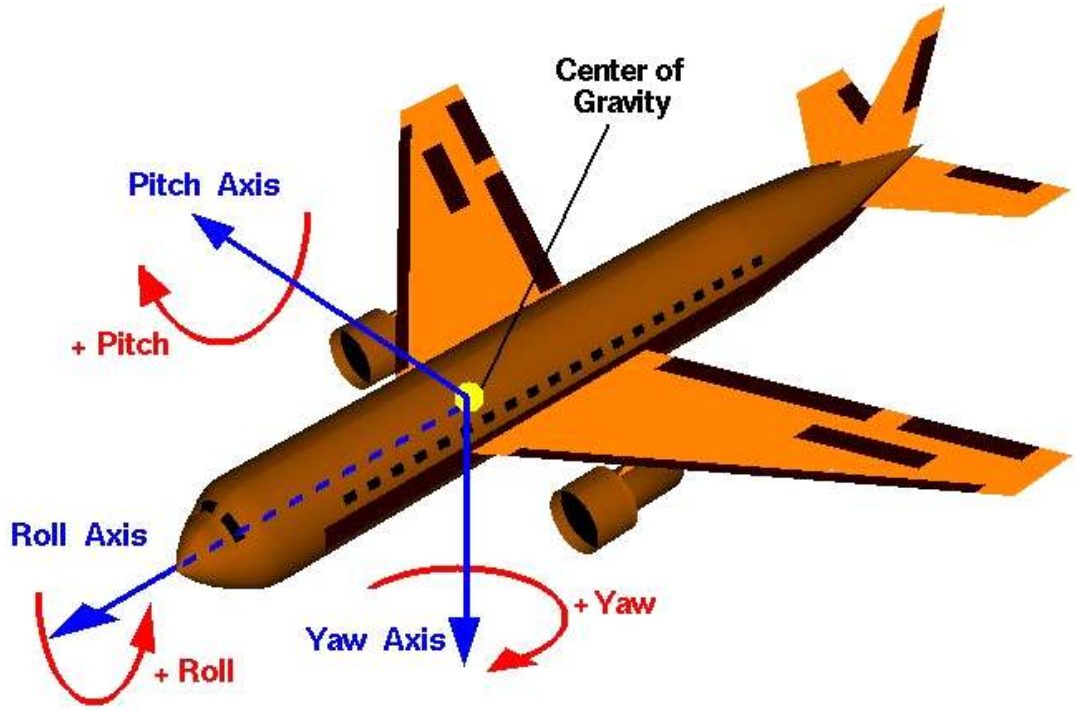


[그림 1] 4족 보행 로봇 1

[그림 2] 4족 보행 로봇 2

2. 항공기의 기본 3축(횡축, 종축, 수직축)

컴퓨터 상이 아닌 일상에서 횡축, 종축, 수직축을 모두 사용하는 경우는 대표적으로 항공기가 있다. 항공기의 회전축인 기본 3축을 바탕으로 이 연구에서 사용할 물체를 기울이는 축을 설정할 수 있다. 횡축은 y축, 종축은 x축, 수직축은 z축에 각각 대응된다.



[그림 3] 항공기의 기본 3축

가. 횡축(피치)

비행기 날개의 좌우 양끝을 연결하는 직선축을 말하며 이 축을 중심으로 상하운동을 한다.

나. 종축(롤)

동체의 앞부분에서 꼬리 부분을 연결하는 직선축을 말하며 이 축을 중심으로 비행기는 좌우운동을 한다.

다. 수직축(요)

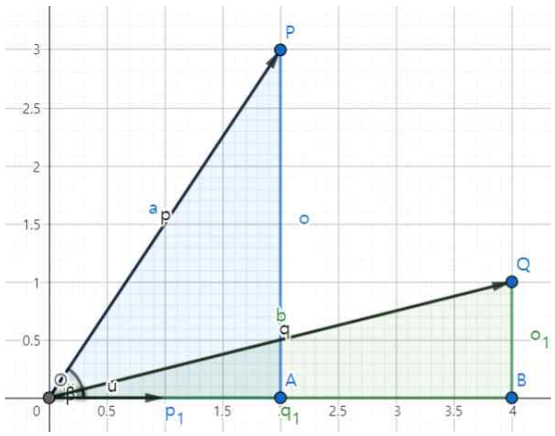
라.

비행기 상부와 하부을 연결하는 직선축을 말하며 이 축을 중심으로 비행기는 좌우운동을 한다. 항공기의 경우처럼 컴퓨터의 3차원 상에서도 3개의 축을 중심으로 회전을 하여 기울기가 정해진다고 가정한다.

3. 수학적 개념의 정의

가. 내적

두 벡터를 표준기저벡터로 나타내었을 때 각 성분끼리의 곱의 합. 영 벡터가 아닌 두 벡터 x, y의 크기 x, y와 x, y가 이루는 각 θ의 cos과의 곱 $xycos\theta$를 x, y의 내적이라 하고, $x \bullet y$ 또는 (x,y)로 나타낸다.



[그림 4] 내적

$\overrightarrow{OP}$, $\overrightarrow{OQ}$가 x축과 이루는 각을 알아보자.

점 P와 Q의 좌표를 $P(a_1,b_1)$, $Q(a_2,b_2)$, x축의 단위벡터를 $\vec{u}=(1,0)$, $\alpha=\angle POA$, $\beta=\angle QOA$라고 할 때 $\cos\alpha=\dfrac{a_1}{\sqrt{a_1^2+b_1^2}}$, $\cos\beta=\dfrac{a_2}{\sqrt{a_2^2+b_2^2}}$이다.

$\theta=\angle POQ$라고 정하고 코사인법칙을 이용하여 $\cos\theta=\cos(\alpha-\beta)=\cos\alpha\cos\beta+\sin\alpha\sin\beta$이므로

$\cos\theta=\dfrac{a_1b_1+a_2b_2}{\sqrt{(a_1^2+b_1^2)}\sqrt{(a_2^2+b_2^2)}}$이다.

나. 법선벡터

직선이나 평면 등에 대해 수직을 이루는 벡터.

공간상의 직선에 대한 법선벡터 - 공간상의 한 직선에 대하여 이 직선을 수직으로 교차하는 무한히 많은 방향을 찾을 수 있다. 그러므로 한 직선에 대한 법선벡터 또한 무한히 많은 방향으로 존재한다. 공간상의 두 직선이 서로 수직을 이루는 조건은 두 직선의 방향벡터의 내적 값이 0이 되는 것이다. 주어진 직선의 방향벡터가 (a,b,c)이고 이 직선의 법선벡터를 (n_1,n_2,n_3)라고 한다면 두 벡터의 내적은

$(a,b,c) \bullet (n_1,n_2,n_3)=an_1+bn_2+cn_3=0$이다.

다. 평면의 방정식

$Ax+By+Cz=D$

라. 세 점을 포함하는 평면의 방정식

세 점 : $P(x_1,y_1,z_1)$, $Q(x_2,y_2,z_2)$, $R(x_3,y_3,z_3)$

평면의 방정식 : $Ax+By+Cz=D$라고 할 때

$\overrightarrow{PQ}\perp\vec{v}$, $\overrightarrow{PR}\perp\vec{v}$이므로 $\overrightarrow{PQ}\cdot\vec{v}=0$, $\overrightarrow{PR}\cdot\vec{v}=0$이다.

III. 연구 방법

1. 2차원 공간 상에서의 3차원 공간 구현

연구에서 사용할 엔트리라는 프로그램은 기본적으로 2차원 공간만을 제공하기 때문에 별도로 2차원 공간에서 3차원 환경을 구현해야 한다. 즉, 사용자가 2차원 공간에서 마우스 커서를 이용해 점의 위치를 선정할 때는 커서 위치인 2차원 상의 좌표 (x_{2D},y_{2D})를 3차원 상의 좌표인 (x_{3D},y_{3D},z_{3D})로 바꿔야 할 필요가 있다. 이 연구에서는 $(x_{3D},y_{3D},z_{3D})=(\frac{2}{3}x_{2D},\frac{1}{3}(x_{2D}+y_{3D}),\frac{2}{3}y_{3D})$로 임의로 할당했다.

2. 프로그램 작동 과정

구현할 프로그램의 작동 과정을 구상한다. 그 과정은 아래와 같다.

가. 세 점 $P(x_1,y_1,z_1)$, $Q(x_2,y_2,z_2)$, $R(x_3,y_3,z_3)$의 좌표를 점 하나씩 마우스로 드래그하여 P,Q,R의 위치를 정한다.

나. 좌푯값을 각각의 리스트에 저장한다. 여기서 리스트란, 현실에서의 목록처럼 여러 개의 값이 모여있는 것을 의미한다. 수는 제한이 없으며 항목을 수정, 삭제, 추가하는 것이 가능하다.

다. 평면의 방정식을 계산한다. 평면의 방정식 : $Ax+By+Cz=D$, 평면의 법선벡터 : $\vec{v}=(A,B,C)$

라. 평면의 법선벡터 $\vec{v}=(A,B,C)$가 z축의 단위벡터 $z(0,0,1)$과 이루는 각 θ를 조사한다.

($\theta=\arccos\frac{A\times0+B\times0+C\times1}{\sqrt{A^2+B^2+C^2}}$)

마. 법선벡터와 z축이 이루는 각은 평면과 지상이 이루는 각과 같으므로 평면이 지상에서 θ만큼 기울어져 있다는 경고를 출력한다.

Ⅳ. 연구 결과

1. 엔트리로 구현한 프로그램의 코드

아래의 그림들은 연구의 결과로 엔트리를 이용해 만들어진 기울기 파악 프로그램의 코드이다.

[그림 5] 엔트리 코드 1

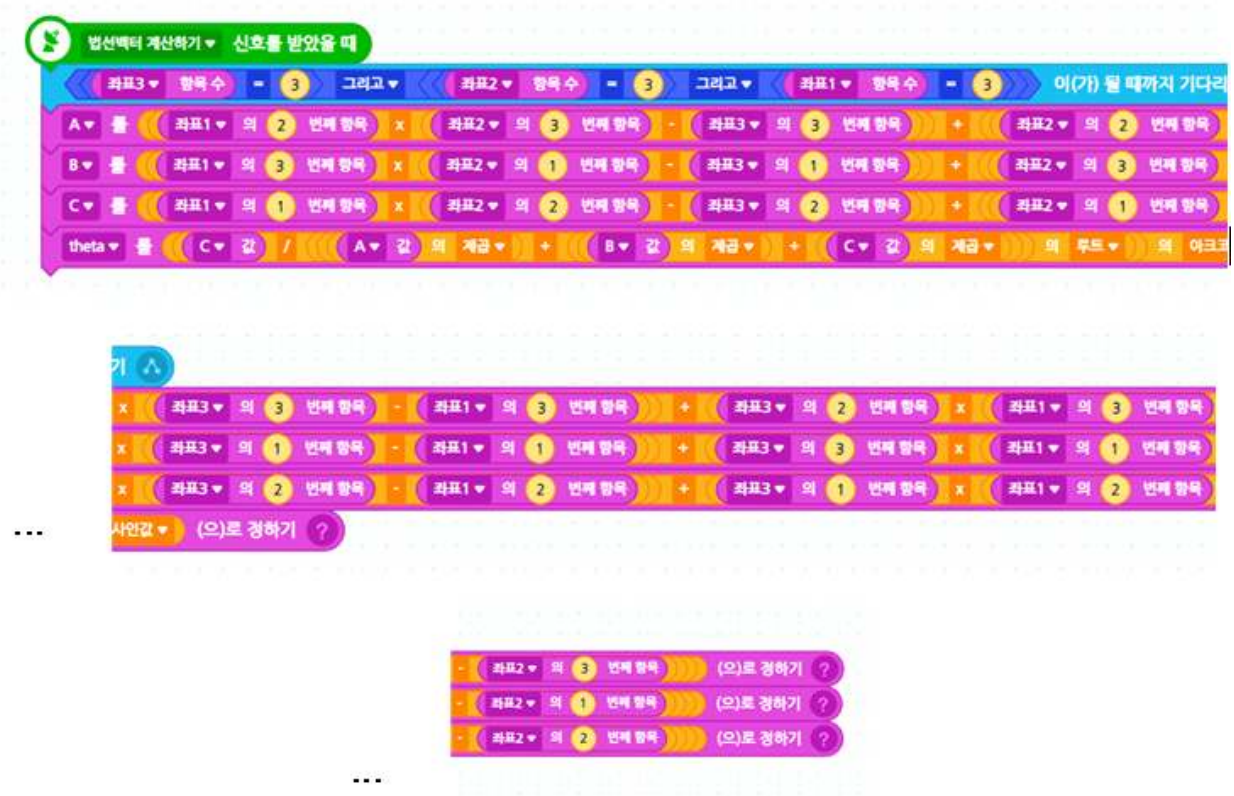
위 코드는 화살표에 입력된 코드다. 모든 점이 옮겨졌을 때, 화살표를 보이게 하고, 평면의 중앙으로 이동시킨 뒤, 계산된 방향을 가리키게 한다. 그 후, 그 값을 출력한다.

[그림 6] 엔트리 코드 2

[그림 7] 엔트리 코드 3

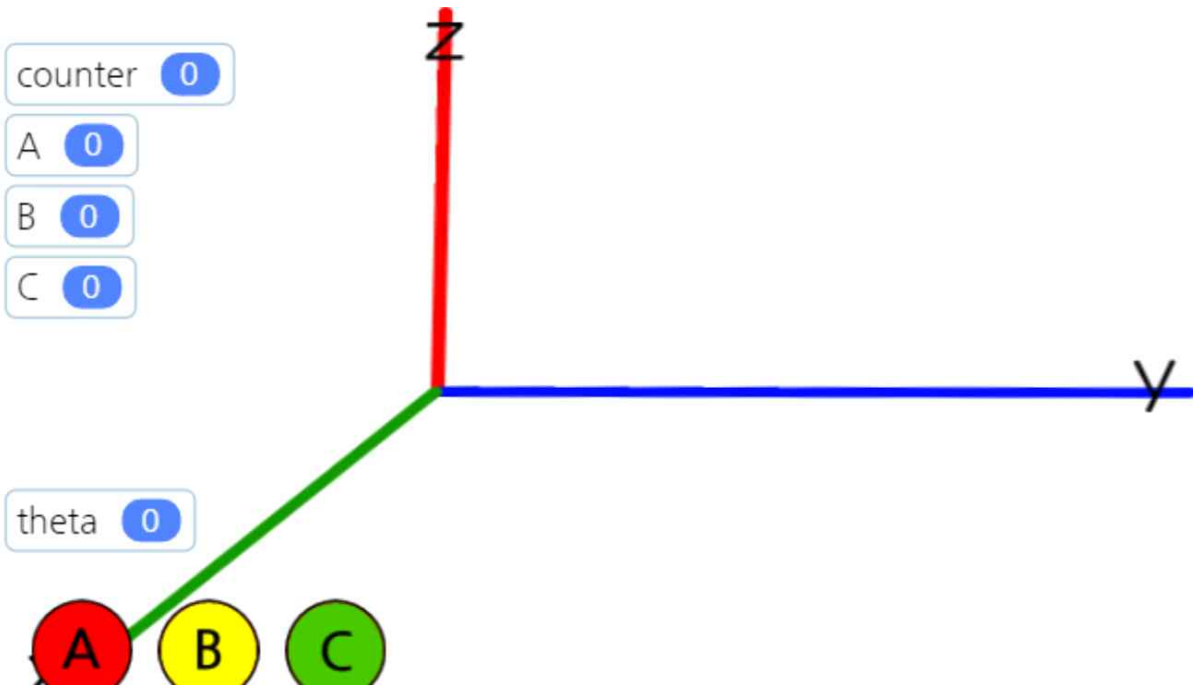
[그림 8] 엔트리 코드 4

위의 세 개의 사진은 점 3개에 각각 입력된 코드다. 마우스로 점을 홀드하고 있을 때는 마우스의 위치로 이동하고, 홀드를 해제하면 그 위치에 점이 고정된다. 그 후, $(x_{3D}, y_{3D}, z_{3D}) = (\frac{2}{3}x_{2D}, \frac{1}{3}(x_{2D}+y_{3D}), \frac{2}{3}y_{3D})$ 에 따라 자신의 위치를 각각의 리스트에 추가한다.

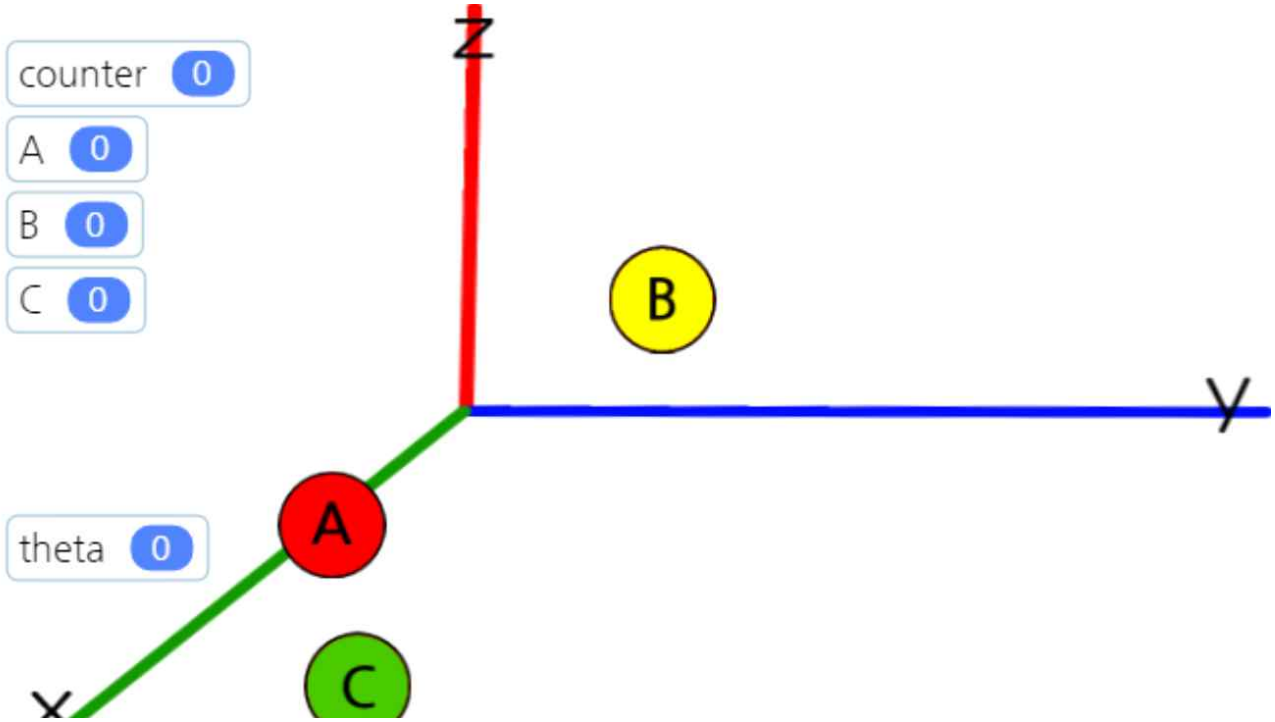


[그림 9] 엔트리 코드 5

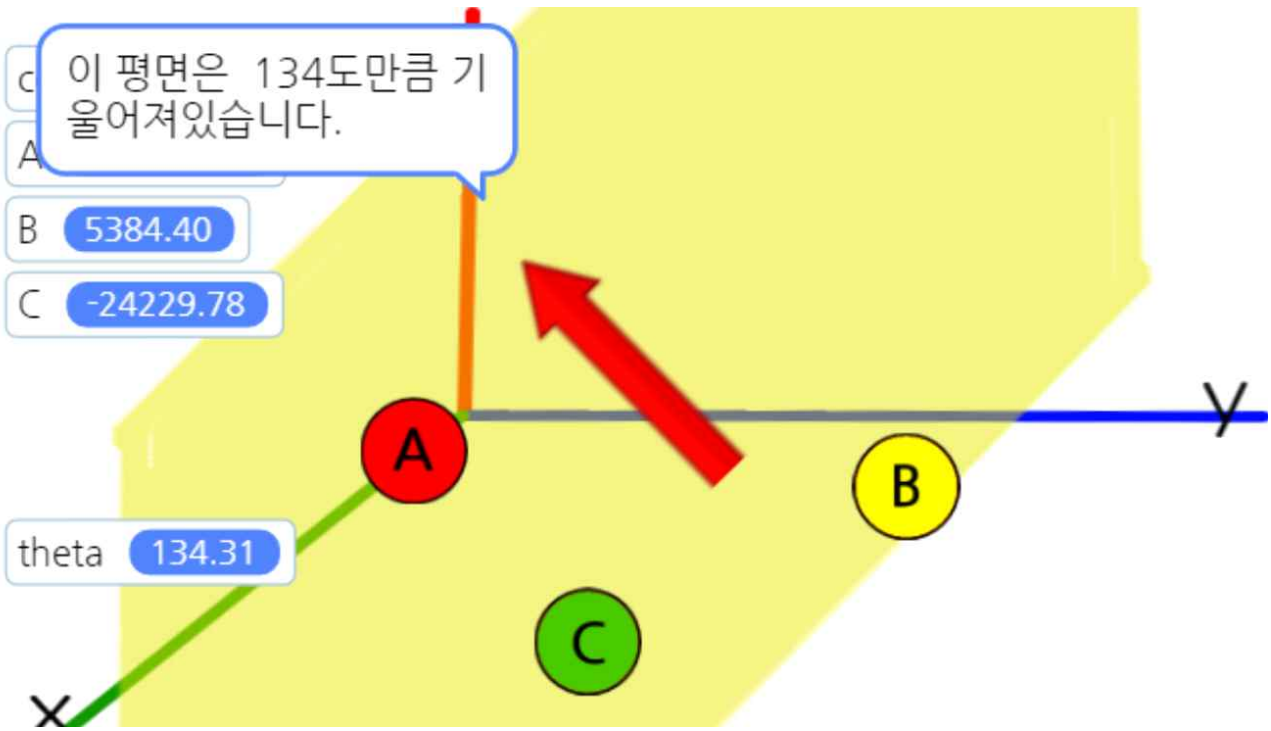
위 코드는 평면에 입력된 코드다. 3개의 점들의 위치가 모두 잡히면 변수 A, B, C 각각 입력한다.



[그림 10] 사용자가 점을 옮기기 전의 상태



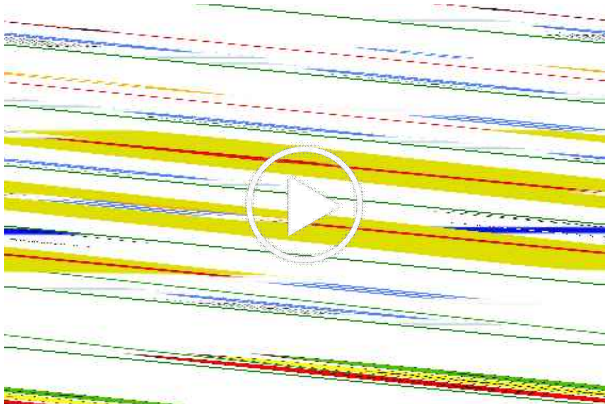
[그림 11] 사용자가 두 점을 옮긴 상태



[그림 12] 사용자가 세 점을 모두 옮기고 결과를 출력한 상태

위 사진은 프로그램을 실행한 즉시 나오는 화면의 사진이다. X축, Y축, Z축이 표현되어 있으며 왼쪽 아래에 있는 세 개의 점은 사용자가 마우스를 이용해 순서에 상관없이 옮길 수 있다. 한 번 자리가 잡힌 점은 고정되어 사용자가 실수로 마우스를 클릭하더라도 움직이지 않는다. 세 점을 모두 옮기면 그 점들로 인해 만들

어진 평면이 기울어진 각도를 출력한다.

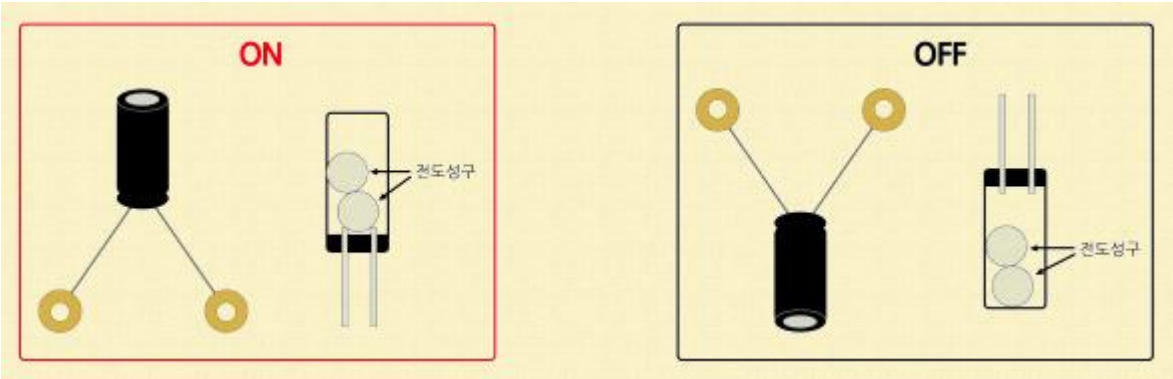


[그림 13(영상 1)] 프로그램의 실행 영상

V. 결론 및 제언

1. 결론

물론 본 연구의 결과로 산출된 프로그램을 쓰지 않더라도 기울기 측정 등과 같은 분야에서 활용할 수 있는 센서는 이미 많이 존재하고 있다. 하지만 비용, 물체의 구조 등의 또 다른 현실적인 문제로 센서의 사용이 제한적일 경우가 있는데, 그럴 경우에 이 프로그램을 이용하면 간단한 벡터의 계산으로도 물체의 기울어진 정도를 알 수 있다. 예를 들어, 건축물의 외벽에 세 개의 위치 센서를 달아 주기적으로 체크하면 건축물이 기울어져 있는지 알 수 있다. 또한, 이 프로그램은 일반적인 센서와는 차별화되는 이점이 있다. 물체의 기울기를 계산하는 센서 중 하나인 아두이노 기울기 센서는 내부에 들어있는 단자가 서로 떨어졌다 붙었다를 반복하면서 작동되거나 진공관 내 수은([그림 12]에서는 '전도성구'라고 표현한다)이 움직이면서 회로가 연결되고 떨어지고를 반복하면서 기울어짐을 알 수 있는 구조이다. 그러나 이 방법으로는 물체가 기울어졌다는 것과 아니라는 것만을 알 수 있고 정확히 얼마나 기울어져 있는지 알 수 없다는 한계점이 존재하기 때문에 이 프로그램은 물체가 기울어 있는 정도의 수치를 계산을 통해 알아낼 수 있다는 부분에서 아두이노 센서 등 다른 기울기 센서와는 차별화되는 의의가 있다.



[그림 14] 아두이노 기울기 센서의 작동 방식

2. 제언

연구 방법의 '2차원 공간 상에서의 3차원 공간 구현' 항목에서 서술했듯 본 연구에서 사용한 프로그램 엔트리는 2차원 공간만을 지원할 뿐, 3차원 공간을 연구자가 직접 구현해야 했다. 때문에 본 연구에서는 (x_{3D}, y_{3D}, z_{3D})를 $(\frac{2}{3}x_{2D}, \frac{1}{3}(x_{2D}+y_{3D}), \frac{2}{3}y_{3D})$같은 임의의 값으로 할당했기 때문에 프로그램의 계산 결과가 실제 3차원 상에서는 다를 수 있다는 한계점이 있다. 또한, 사용자가 점들을 마우스로 드래그해 특정 위치에 놓을 때에도 사용자가 원하는 정확한 위치에 점을 놓을 수 없다는 한계점도 있다. 이 문제는 사용자가 각 점들의 x좌표, y좌표, z좌표를 직접 입력할 수 있도록 해주는 기능을 추가하는 방법으로 해결할 수 있다. 아니면 실제 3차원 공간처럼 사용자가 카메라의 시점을 직접 움직일 수 있도록 해주는 기능을 추가하는 방법도 있는데, 이 경우는 2차원 상에서 3차원 상의 이동까지 구현을 해야 하므로 시간과 노력을 심각하게 많이 들여야 할 수도 있다. 이를 통해 2차원 공간만 제대로 구현 가능한 엔트리보다는 유니티(Unity) 등 3차원 공간을 기본적으로 제공하는 프로그램을 사용해야 좀 더 의미 있는 결과를 얻을 수 있다는 생각이 들었다. 마지막으로, 이 프로그램에는 각도 계산 뿐만이 아니라 새로운 기능을 더하면 더 사용성, 활용성이 높아질 것 같다. 물리 엔진을 적용해 바닥 평면에 한 평면을 놓은 후, 바닥 평면을 움직이면 그 평면이 얼마나 기울어지는지 같은, 기존 기울기 센서에서는 접근할 수 없고 명확히 차별화되는 새로운 기능이 필요할 것 같다.

참 고 문 헌

Aircraft Rotations. (2022.07.21.). NASA Gleen Research Center. 검색일 : 2023.05.05
URL : https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/aircraft-rotations/

ProgC.(2013.12.6.).디스이즈게임. 검색일 : 2023.05.05.
URL : https://m.thisisgame.com/hs/nboard/212/?series=99&n=51407

기울기 센서로 LED 제어하기. (2018.01.11.). Kocoafab. 검색일 : 2023.05.05.
URL : https://kocoafab.cc/tutorial/view/735

곽노식, 김정엽.(2020).4족 보행 로봇의 자세 안정화 전략.로봇과 인간,17(2),11-17.

나소연.(2009).베일이 들려주는 벡터 이야기.

“내적” (표기없음). Doopedia, 검색일 : 2023.04.03. URL :
https://www.doopedia.co.kr/doopedia/master/master.do?_method=view&MAS_IDX=101013000755051

박찬.(2022.05.11.).AI타임즈. 검색일 : 2023.05.05. URL:
https://www.aitimes.com/news/articleView.html?idxno=144527

“법선벡터” (표기없음). Doopedia, 검색일 : 2023.04.03. URL :
https://www.doopedia.co.kr/doopedia/master/master.do?_method=view&MAS_IDX=181119001596742

“자이로스코프” (표기없음). Doopedia, 검색일 : 2023.05.05. URL :
https://www.doopedia.co.kr/doopedia/master/master.do?_method=view&MAS_IDX=101013000857435

“항공역학” (표기없음). Doopedia, 검색일 : 2023.05.05. URL :
https://www.doopedia.co.kr/doopedia/master/master.do?_method=view&MAS_IDX=101013000867572

“휴머노이드 로봇” (표기없음). Doopedia, 검색일 : 2023.05.05. URL :
https://www.doopedia.co.kr/doopedia/master/master.do?_method=view&MAS_IDX=101013001000035