
드론 코딩 체험북

v2.0

헬로앱스

<http://www.helloapps.co.kr>

02

다양한 형태의 모양 만들기

학습 목표

- 다양한 형태의 모양 만들기 명령어를 학습할 수 있다.
- 모양과 모양을 서로 결합하여 새로운 형태의 모양을 창조할 수 있다.

실습 개요

- 각각의 모양을 만드는 명령어를 활용해 본다.
- 크기와 위치를 수정해 본다.
- 물체를 회전시켜 본다.
- 부모와 자식 관계를 이해한다.
- 결합된 형태의 새로운 모양을 생성해 본다.

준비물

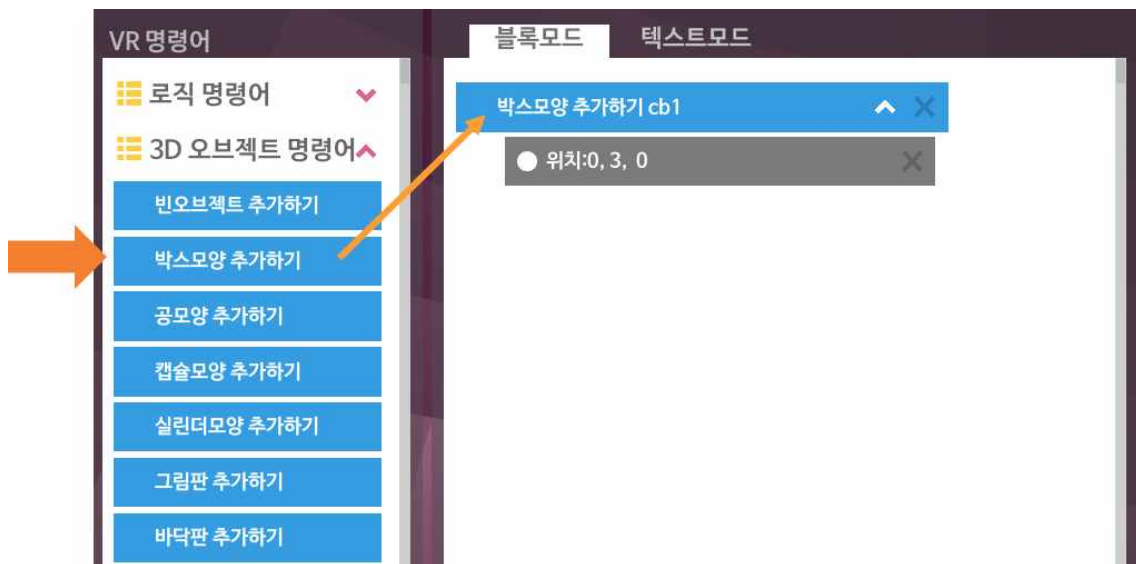
- VR 코딩 앱 (안드로이드용 또는 PC용, 설치 과정 참조)
- 스마트폰 (안드로이드 기기) 또는 PC (윈도우 OS)



2.1 물체 회전 시키기

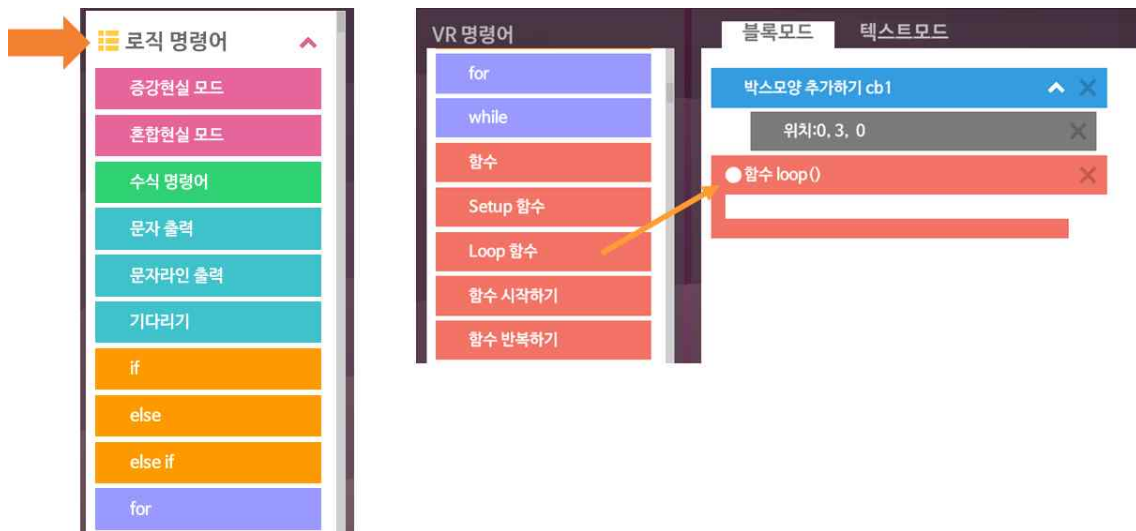
회전축 박스 모양 추가하기

- 새로운 파일 작성 후, 박스모양 추가하기 명령어를 추가한 후, 크기를 다음과 같이 지정한다.



loop 함수 추가하기

- 로직 명령어에 있는 loop 함수를 프로그램 아래에 추가한다.



모양 회전시키기

- 도구 명령어 항목에 있는 회전시키기 명령어를 loop 함수 내에 추가한다.



● cb1 회전시키기(0, 1, 0)



- 실행하여 결과를 확인해 본다.



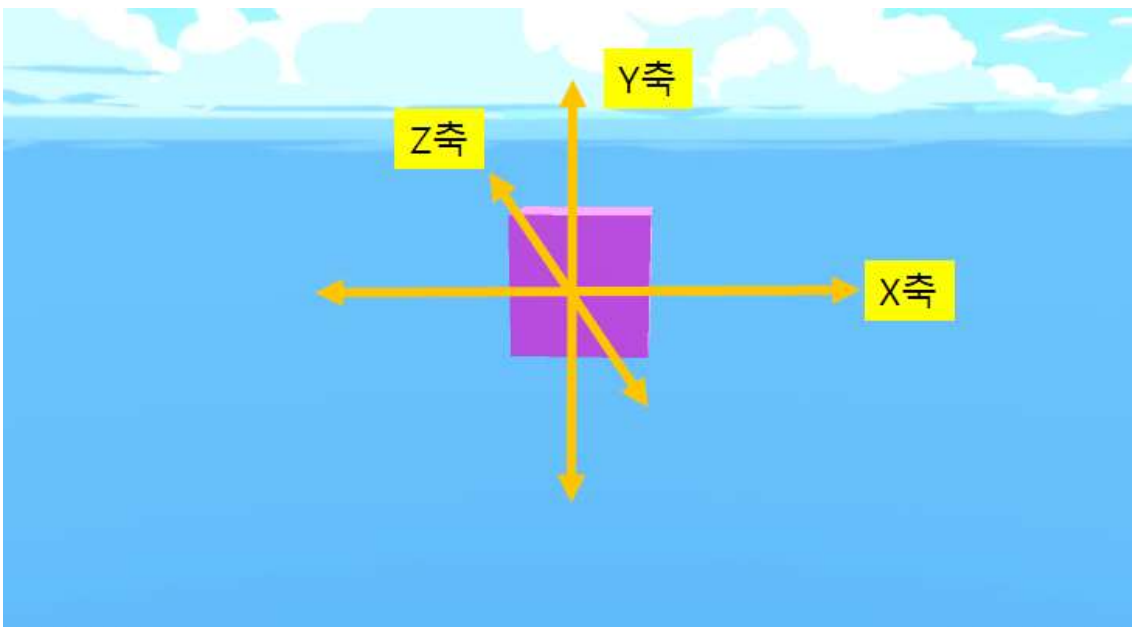
회전 속도 및 방향 변화

- 박스의 회전 속도를 변화 시켜 본다.
- 박스의 회전 값을 마이너스 (-) 값으로 변화시키면 회전 방향이 어떻게 변화되는 지 관찰한다.

● cb1 회전시키기(0, -1, 0)

회전축의 변화

- 아래 그림을 보고 현재 박스가 어느 축을 기준으로 회전하는 지 확인한다.



실습

- 박스를 X축으로 회전 시키기 위해서는 어느 축의 값을 지정해 주어야 하는 지 생각해 보고 값을 수정해 보자.
- 박스를 Z축으로 회전 시키기 위해서는 어느 축의 값을 지정해 주어야 하는 지 생각해 보고 값을 수정해 보자.
- X, Y, Z 축에 동시에 회전 값을 주었을 경우 어떠한 결과가 발생하는 지 예측해 보고 프로그램을 실행해 본다.

2.2 물체를 부모와 자식으로 연결시키기

Z축으로 회전시키기

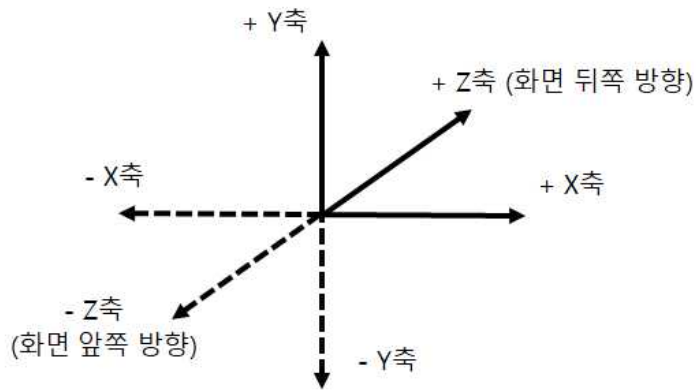
- 기존에 작성한 코드에서 기존 박스를 Z축으로 2의 속도로 회전하도록 loop 함수에 있는 회전시키기 명령어의 값을 다음과 같이 수정한다.



- 실행 결과를 확인해 본다.

날개 모양 추가하기

- 현재 생성된 박스외에 추가로 가로로 긴 날개 모양의 박스를 추가해 본다. 아래 그림을 참조하여 가로 방향으로 길게 만들려면 어떠한 축의 값을 크게 해야 할지 예측해 본다.



● 박스모양 추가하기 cb1

위치:0, 3, 0

박스모양 추가하기 cb2

위치:0, 3, 0

크기:5, 0.5, 0.5

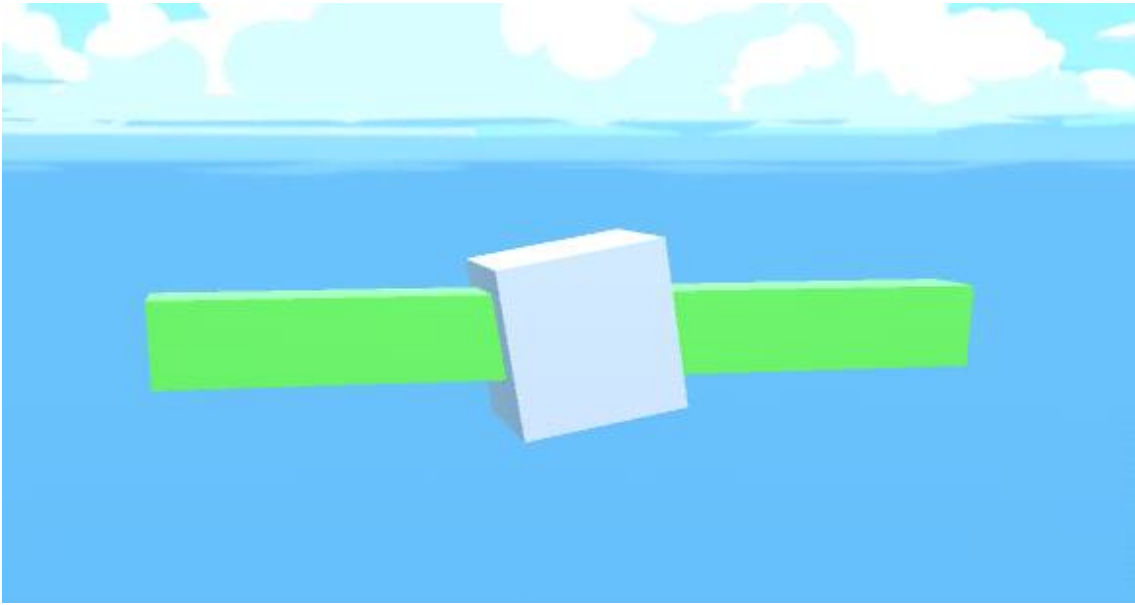
함수 loop()

cb1 회전시키기(0, 0, 2)

- 새로 추가한 박스의 이름은 cb2 로 수정하여 기존 박스의 이름과 겹치지

않도록 한다.

- 실행하여 결과를 확인해 본다.



문제점

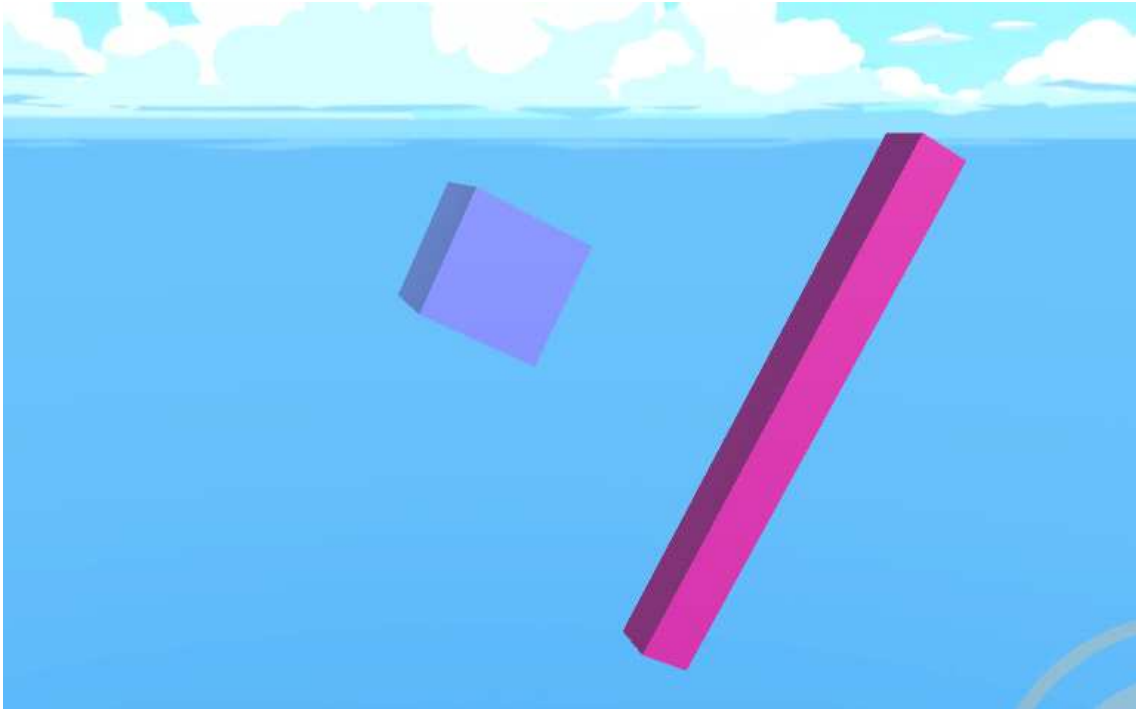
- 새로 추가한 날개 모양의 박스(cb2)는 그대로 있고 가운데의 박스만 회전하는 결과를 보게 된다.
- 새로 추가한 날개 모양의 박스(cb2)와 가운데 있는 cb1 박스가 같이 회전하도록 하려면 cb2 박스와 cb1 박스를 하나의 그룹으로 묶어주는 과정이 필요하다.

부모와 자식 관계로 연결하기

- 새로 추가한 날개 모양의 박스(cb2)를 cb1 박스와 하나의 그룹으로 묶어 주기 위해서는 c2를 c1의 자식으로 연결시켜 주어야 한다.
- cb2의 옵션중에서 부모오브젝트 옵션을 추가한 후, 부모 이름을 cb1으로 지정해 준다.



- 실행하여 결과를 확인해 본다.



문제점

- 기대한 바와 달리 cb2 박스가 cb1으로부터 떨어져서 연결되어 있는 것을 볼 수 있다.
- 이러한 결과가 나온 이유는 부모오브젝트 옵션이 사용된 경우, 위치 옵션 값은 부모의 위치로부터 상대 위치 값으로 인식되기 때문이다.



부모 cb1의 중심 위치로
부터의 상대적인 위치

- cb2의 위치 옵션을 제거하고 다시 실행해 본다.

박스모양 추가하기 cb1

위치:0, 3, 0

● 박스모양 추가하기 cb2

크기:5, 0.5, 0.5

부모오브젝트:cb1

함수 loop()

cb1 회전시키기(0, 0, 2)

- 이제 기대한 바와 같이 cb1과 cb2 박스가 같이 회전하고, cb2는 cb1의 중심과 일치하도록 위치가 수정된 것을 볼 수 있다.



두 번째 날개 추가하기

- cb3 박스를 추가하고 다음과 같이 Y축으로 긴 막대가 되도록 옵션을 지정해 준다.

● 박스모양 추가하기 cb1 ^ X

위치:0, 3, 0 X

박스모양 추가하기 cb2 ^ X

크기:5, 0.5, 0.5 X

부모오브젝트:cb1 X

박스모양 추가하기 cb3 ^ X

크기:0.5, 5, 0.5 X

부모오브젝트:cb1 X

함수 loop() X

cb1 회전시키기(0, 0, 2) X

- 실행하여 결과를 확인해 본다.

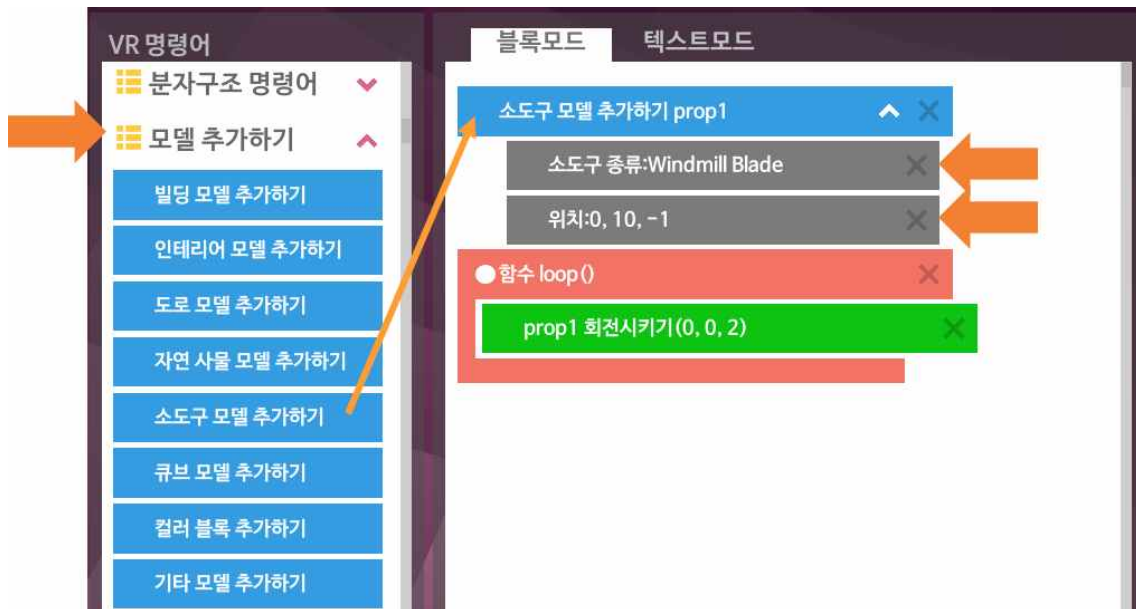


- 풍차의 회전 속도 및 방향을 수정해 본다.

2.3 3D 모델링 추가하기

풍차 모양의 모델링 추가하기

- 앞서 제작한 풍차의 날개 대신에 미리 만들어져 있는 프로펠러를 추가해 본다.
- VR명령어 중, 맨 아래에 있는 모델 추가하기 그룹을 선택하면, 소도구 모델 추가하기 명령어를 볼 수 있다. 소도구 모델 추가하기 명령어를 추가한 후, 소도구 종류는 Windmill Blade, 위치는 아래와 같이 지정해 준다.



- 맨 아래 회전시키기 명령어의 값도 기존 cb1에서 prop1으로 수정해 준다.

prop1 회전시키기(0, 0, 2) X

- 실행하여 결과를 확인해 보자. 풍차 날개가 너무 높이 있어서 화면에 보이지 않는 문제가 발생한다.



카메라 위치 수정하기

- 풍차 날개가 처음에 바로 보이도록 카메라의 위치를 다음과 같이 수정한다.

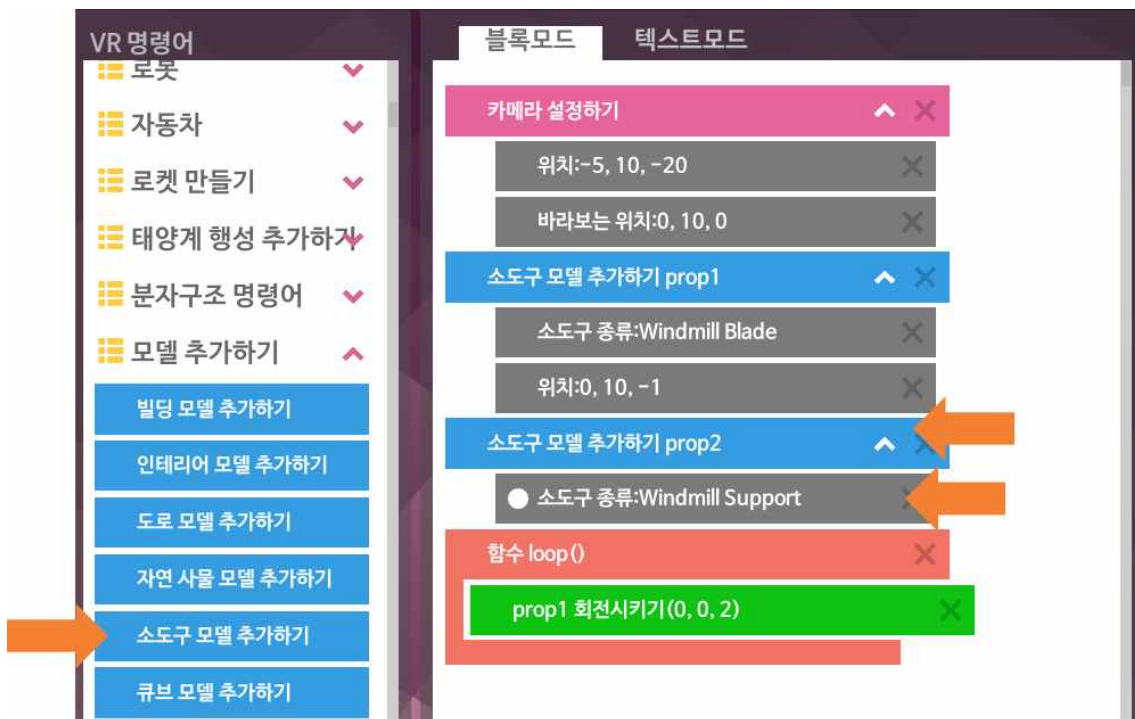


- 실행하여 결과를 확인해 보면 풍차 날개가 화면 가운데에 보이는 것을 알 수 있다.



풍차 기둥 추가하기

- 소도구 모델링 추가하기를 통하여 풍차 기둥을 다음과 같이 추가해 준다.



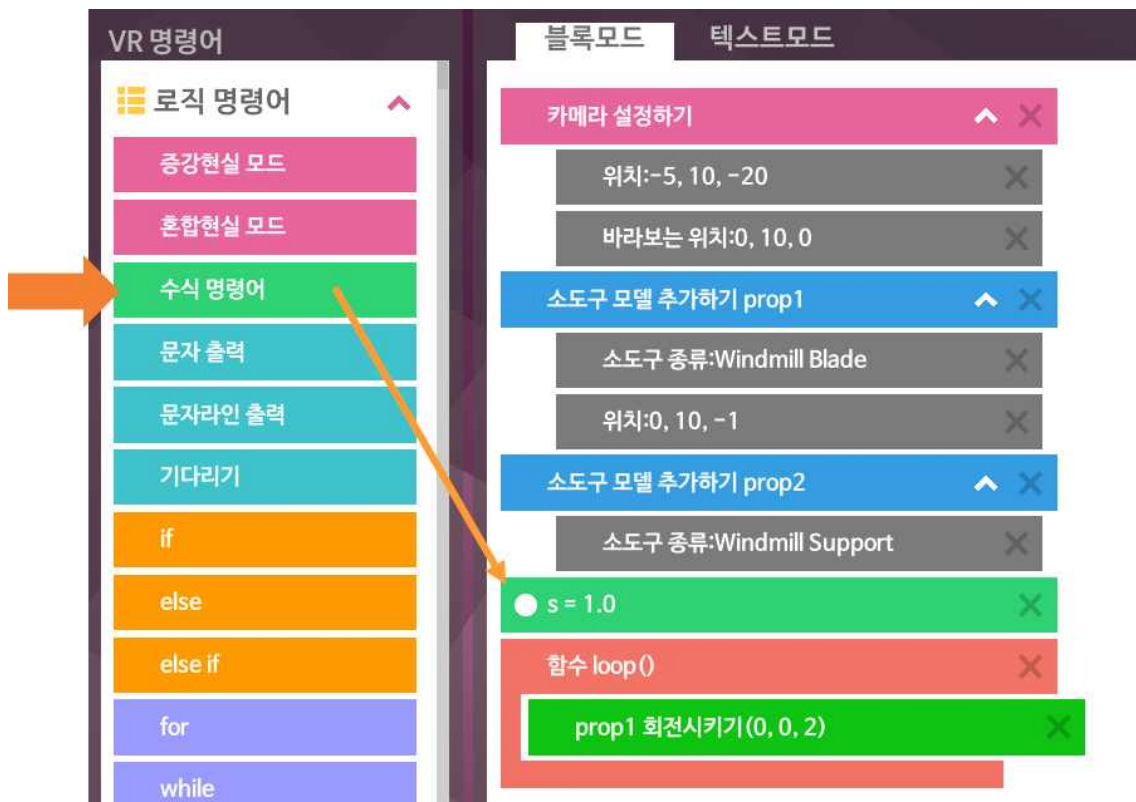
- 소도구 모델 추가하기 명령어의 이름이 겹치지 않도록 주의한다.



2.4 변수 값으로 회전 속도 제어하기

변수 사용하기

- 앞서 제작한 풍차의 회전 속도를 변수로 제어해 본다.
- 수식 명령어를 loop 함수 바깥에 추가한 후, 다음과 같이 수식을 입력해 본다.



- 수식 명령어에는 다음의 수식을 입력해 준다.

$$s = 1.0$$

- prop1 회전시키기 명령어에 변수 s를 입력해 준다.



- 프로그램을 실행시켜 결과를 확인해 본다.
- 수식 명령어의 변수 값을 수정하여 풍차의 회전 속도와 방향을 수정해 본다.

2.5 점진적으로 회전 속도 증가시키기

수식 명령어 사용하기

- 앞서 정의한 변수 값을 점진적으로 증가시켜 본다. 먼저, 변수 s 의 초기값을 0으로 수정한다.

```
s = 0
함수 loop()
prop1 회전시키기(0, 0, s)
```

- loop 함수 맨 위에 새로운 수식 명령어를 추가한 후, 다음과 같은 수식을 입력해 준다.

$s = s + 0.1$

```
s = 0
함수 loop()
s = s + 0.1
prop1 회전시키기(0, 0, s)
```

- 실행하여 결과를 확인해 본다.

카메라 설정하기

^ X

위치:-5, 10, -20

바라보는 위치:0, 10, 0

소도구 모델 추가하기 prop1

^ X

소도구 종류:Windmill Blade

위치:0, 10, -1

소도구 모델 추가하기 prop2

^ X

소도구 종류:Windmill Support

s = 0

X

함수 loop()

X

s = s + 0.1

X

prop1 회전시키기(0, 0, s)

X

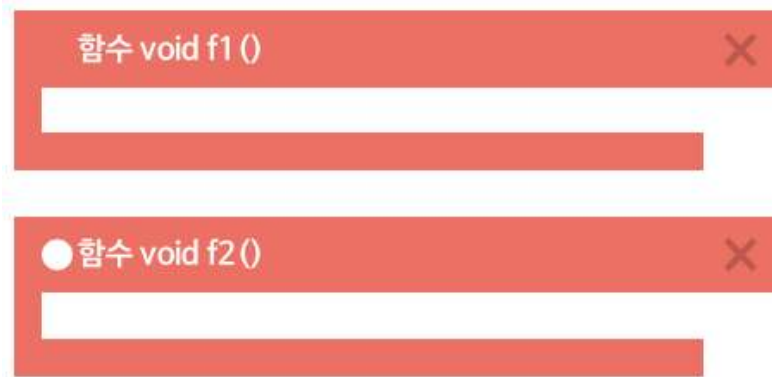
2.6 속도 조절 장치 만들기

함수 만들기

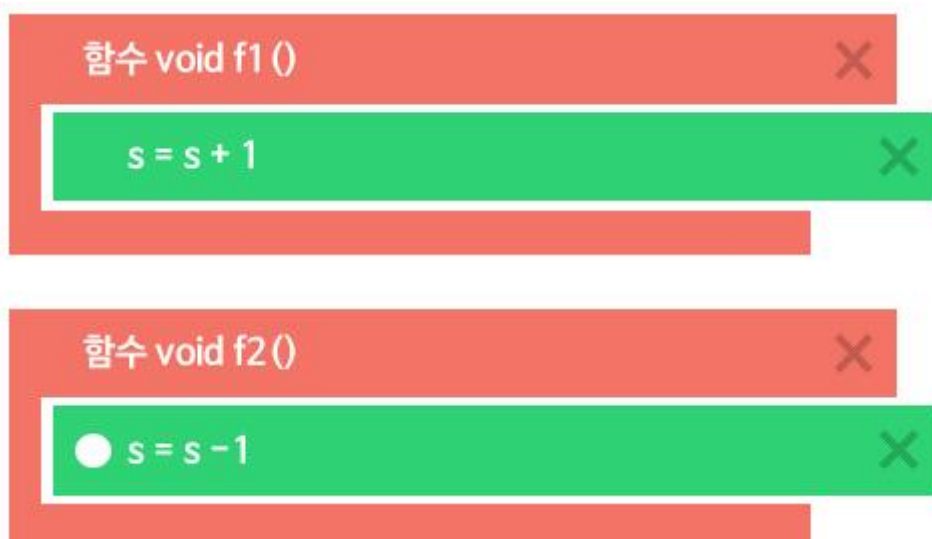
- 조이스틱 버튼을 이용하여 풍차의 회전 속도를 조절하는 장치를 만들려고 한다.
- 조이스틱 버튼을 클릭하면 함수가 호출되는데, 이 때 호출되는 함수를 미리 만들어 놓아야 한다. 다음과 같이 로직 명령어 그룹에 있는 “함수” 명령어를 프로그램 맨 아래에 추가해 준다.



- “함수” 명령어를 한번 더 추가하여 프로그램 맨 아래에 추가해 준다. 함수 f1()과 f2()가 생성되었다.

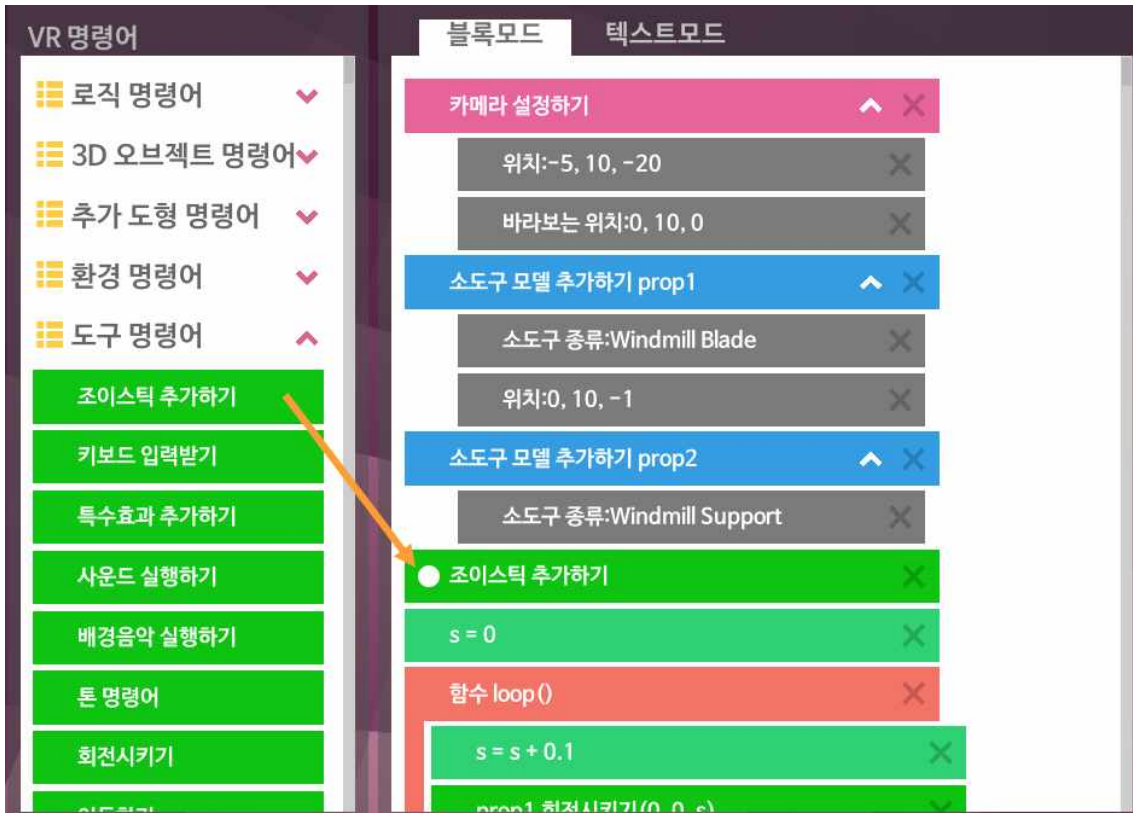


- 나중에 함수 f1()은 회전 속도를 증가시키기 위한 목적으로 사용하고, 함수 f2()는 회전 속도를 감소시키기 위한 목적으로 사용할 예정이다.
- 함수 f1() 안에 수식 명령어를 추가한 후, 다음과 같이 전역변수 s의 값을 1씩 증가시키는 명령어를 추가한다. 그리고 함수 f2()에도 수식 명령어를 추가한 후, 전역변수 s의 값을 1씩 감소시키는 명령어를 입력한다.

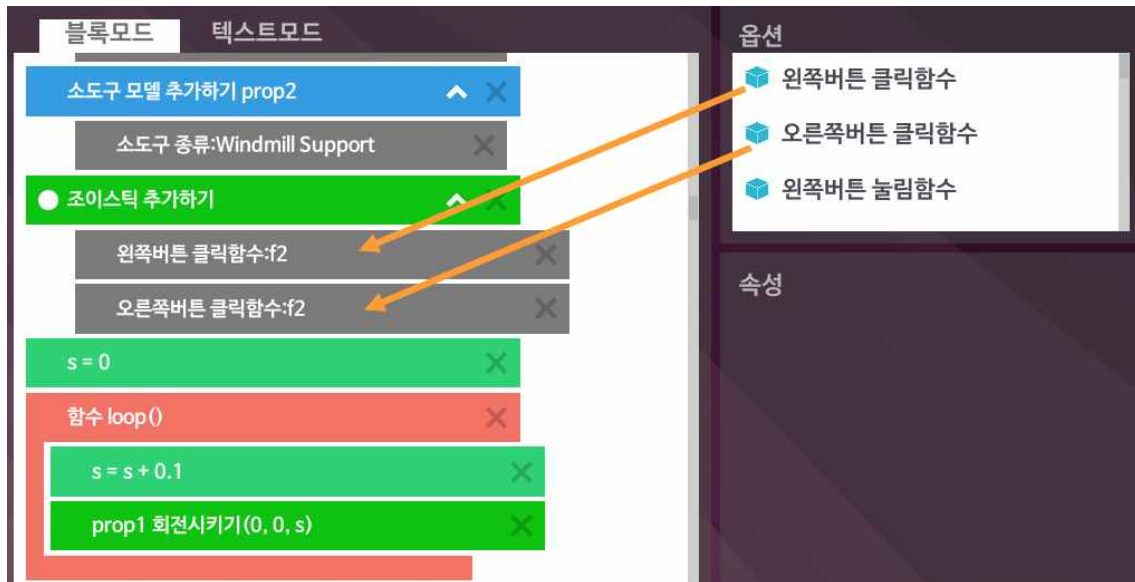


조이스틱 명령어 추가하기

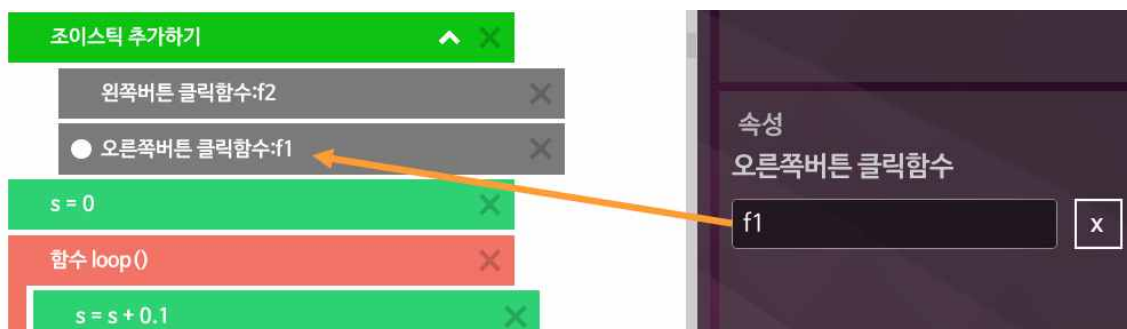
- 도구 명령어 그룹의 맨 앞에 있는 “조이스틱 추가하기” 명령어를 loop 함수 위쪽에 다음과 같이 추가해 준다.



- 조이스틱 추가하기 명령어의 옵션에서 왼쪽버튼 클릭함수 옵션과 오른쪽버튼 클릭함수 옵션을 추가한다.



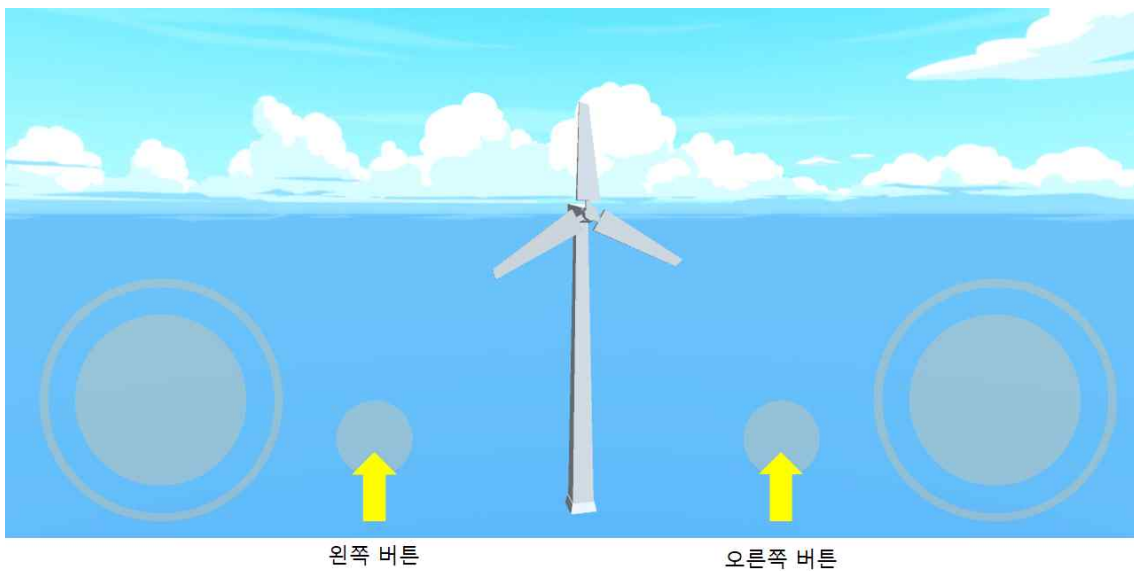
- 옵션에서 값은 다음과 같이 왼쪽 함수에는 함수이름 f2, 오른쪽 함수에는 함수이름 f1을 입력해 준다.



- 기존 loop 함수 안에 있는 s 값을 증가시켜 주는 명령어를 삭제한다.

```
s = 0
함수 loop()
  ● prop1 회전시키기(0, 0, s)
```

- 프로그램을 실행한 후 다음과 같이 화면 상의 작은 버튼을 클릭해 준다.



- 전체 코드는 다음과 같다.

카메라 설정하기 ^ x

위치:-5, 10, -20

바라보는 위치:0, 10, 0

소도구 모델 추가하기 prop1 ^ x

소도구 종류:Windmill Blade

위치:0, 10, -1

소도구 모델 추가하기 prop2 ^ x

소도구 종류:Windmill Support

조이스틱 추가하기 ^ x

왼쪽버튼 클릭함수:f2

오른쪽버튼 클릭함수:f1

s = 0 x

함수 loop() x

prop1 회전시키기(0, 0, s) x

함수 void f1() x

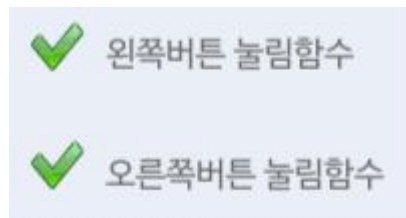
s = s + 1 x

함수 void f2() x

s = s - 1 x

실습

- 조이스틱 버튼을 클릭할 때 마다, 문자라인 출력 명령어를 이용하여 화면에 현재 상태의 속도를 표시하는 기능을 추가하시오.
- 조이스틱 명령어의 옵션 중에서 기존 클릭함수 대신에, 왼쪽버튼 눌림함수, 오른쪽버튼 눌림함수를 사용하면 결과가 어떻게 달라지는 지 확인하시오.



- 눌림함수는 버튼을 계속 누르고 있으면 해당 함수가 계속 호출된다.

2.7 나만의 회전 모양 만들기

창작 실습 1

- 다양한 형태의 도형 명령어와 부모-자식 관계를 이용하여 나만의 창의적인 풍차 또는 선풍기 등 다양한 회전 창작물을 설계하고 만들어 본다.

창작 실습 2

- 헬리콥터처럼 Y축을 중심으로 회전하는 회전 날개 장치를 설계하고 구현해 본다.