

## 05 태양계 모형 만들어 보기

### 학습 목표

- 3D 오브젝트 물체를 부모와 자식으로 연결하여 제어할 수 있다.
- 물체의 회전과 조명을 다룰 수 있다.

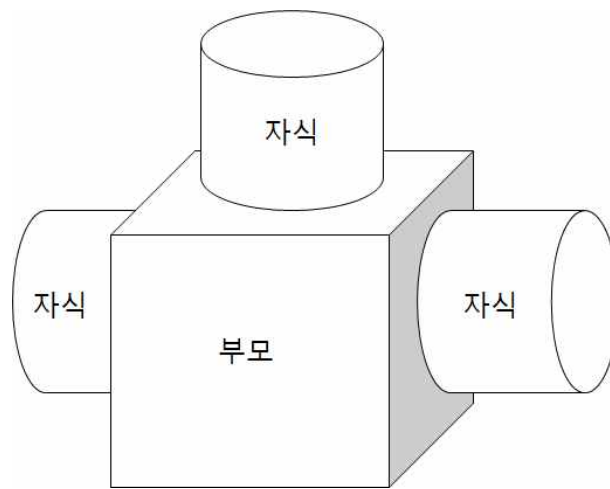
### 실습 개요

- 부모와 자식으로 연결된 복합 형태의 물체를 개발해 본다.
- 생성된 물체를 회전시켜 본다.
- 조명을 제어해 본다.
- 태양계 및 지구와 달 환경을 구성하고 공전과 자전을 구현해 본다.

## 5.1 여러 모양이 합쳐진 물체 만들기

### 여러 물체를 하나로 연결하기

- 가상현실 코딩에서 여러 물체를 하나로 연결하기 위해서는 물체들을 부모와 자식 관계로 연결해 주어야 한다.



- 위의 그림은 3개의 자식 실린더 모양이 하나의 부모 상자 모양에 연결되어 있는 형태를 보여준다.
- 여러 물체를 부모와 자식 관계로 연결하면 좋은 점은 부모 객체와 자식 객체들이 하나처럼 움직이게 된다는 것이다.
- 복잡한 형태의 모양들은 단순한 도형들을 서로 결합하여 만들 수 있다.

## 눈사람 만들기

- 2개의 공 모양 오브젝트를 생성한 다음 2개를 부모와 자식으로 연결하여 눈 사람 형태의 복합 오브젝트를 만들어 보자.
- 먼저 바닥판을 추가한 후, 공 모양을 2개 추가하여 눈사람 형태가 되도록 한다.

바닥판 추가하기 plane1

공모양 추가하기 sphere1

위치:0 1 0

크기:2 2 2

질량:1.0

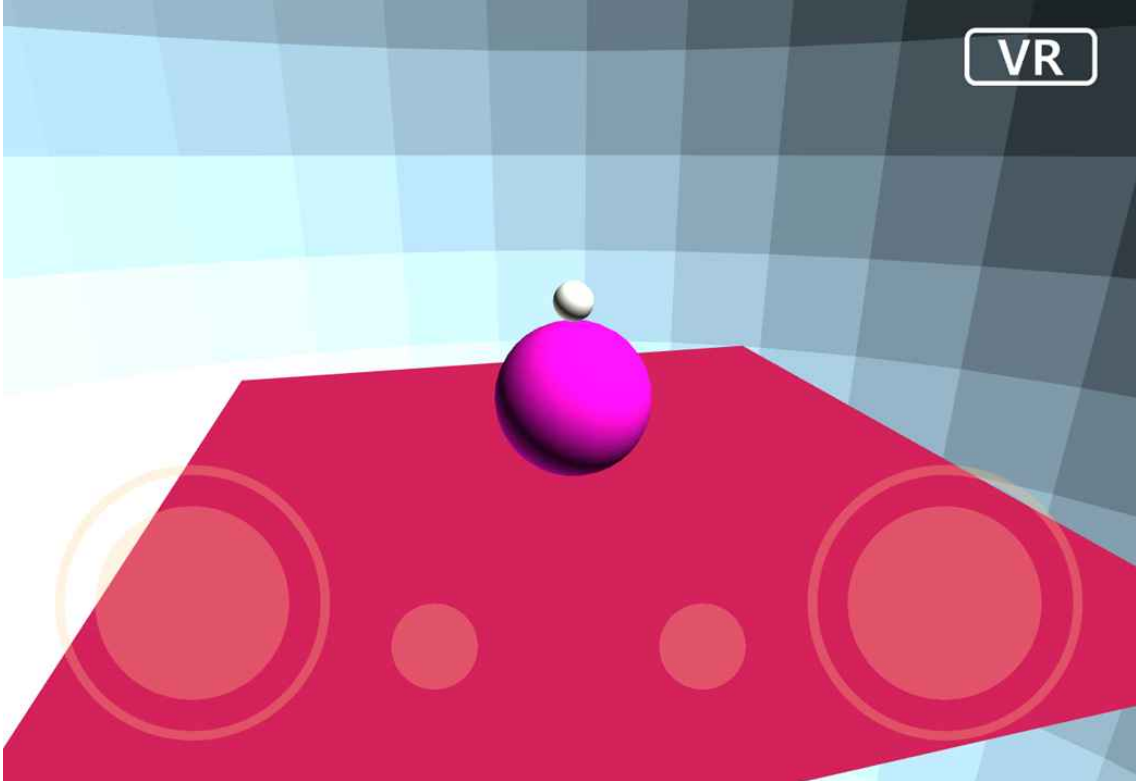
공모양 추가하기 sphere2

위치:0 2.25 0

크기:0.5 0.5 0.5

질량:1.0

- 실행 결과는 다음과 같다.

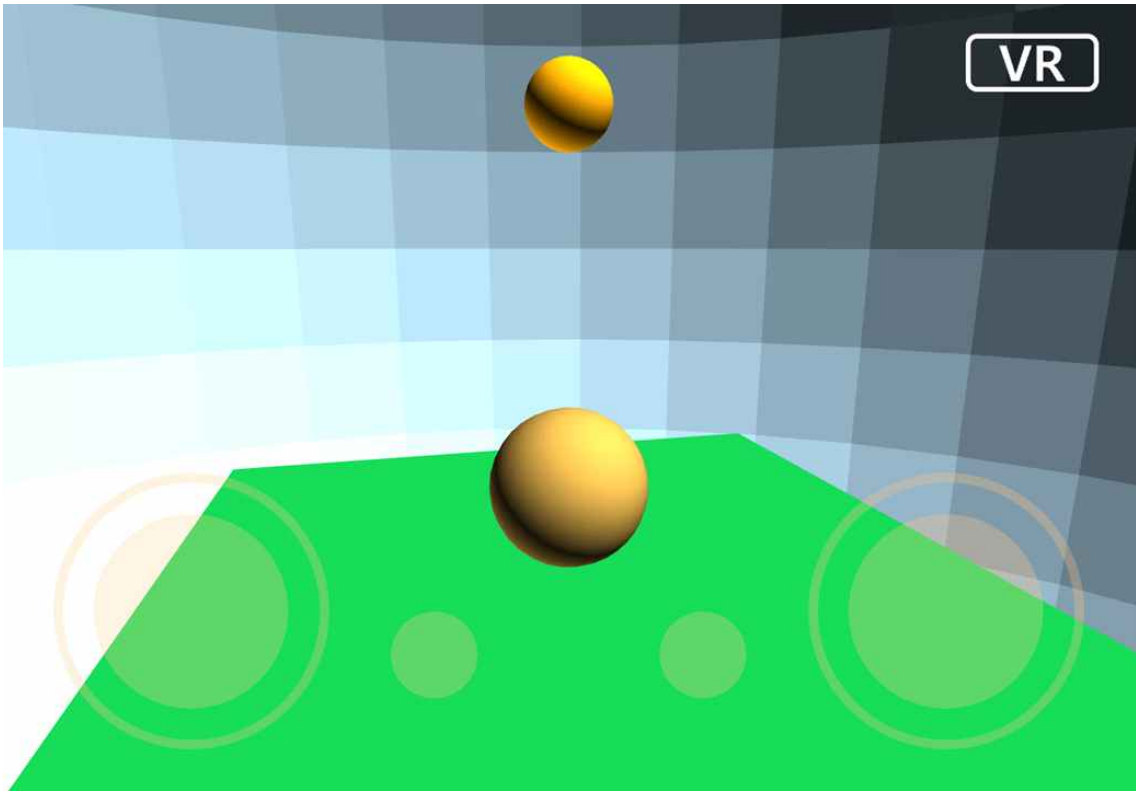


## 부모와 자식으로 연결하기

- 두 번째 공(sphere2)의 옵션에 부모오브젝트를 추가하고 부모의 이름을 sphere1으로 지정해 준다.



- 위의 코드를 실행해 보면 이전 실행과 전혀 다른 결과가 보이게 된다.



- 부모오브젝트 옵션이 추가되면, 자식 오브젝트에 있는 위치나 크기 값들은 부모 오브젝트의 위치나 크기 값을 기준으로 상대적인 값으로 계산된다. 즉, 자식 오브젝트의 위치도 좌표 중심으로 부터의 위치가 아니라 부모오브젝트의 중심 위치로 부터의 상대적인 거리를 의미하게 된다.
- 위의 코드에서 위쪽의 작은 공은 거리가 아래 쪽의 부모오브젝트로부터 상대적인 거리를 가져야 하므로 다음과 같이 코드를 수정해 보자.

바닥판 추가하기 plane1

공모양 추가하기 sphere1

위치:0 1 0

크기:2 2 2

질량:1.0

공모양 추가하기 sphere2

위치:0 0.75 0

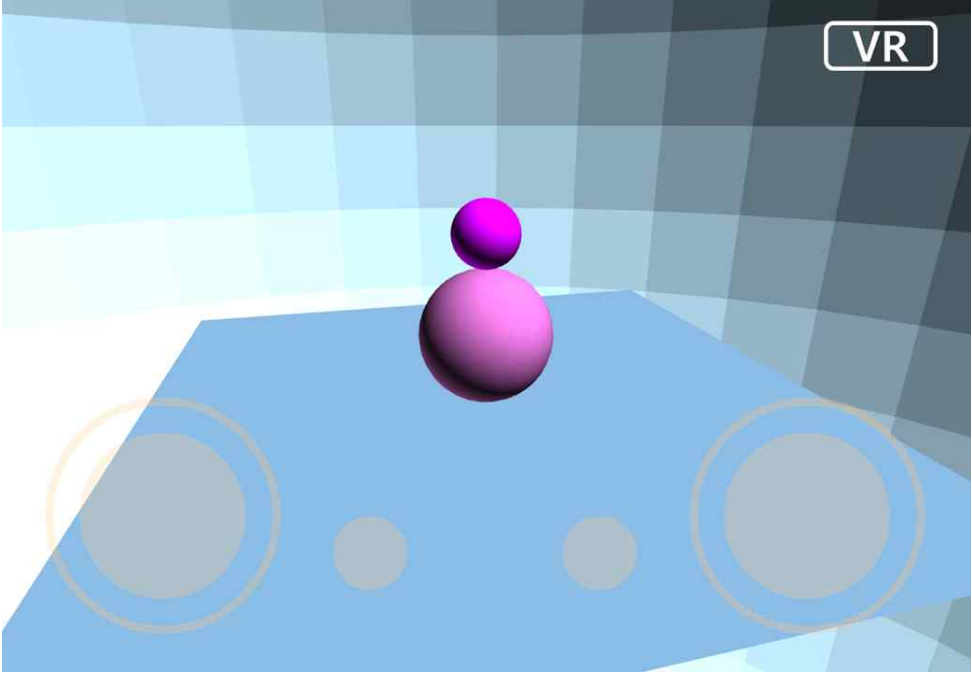
크기:0.5 0.5 0.5

질량:1.0

부모오브젝트:sphere1

위치

|   |   |      |   |
|---|---|------|---|
| X | - | 0    | + |
| Y | - | 0.75 | + |
| Z | - | 0    | + |



## 실습

---

- ▶ 자신만의 눈사람을 창의적으로 만들어 보자.
- ▶ 눈사람 외에 로봇과 같은 다른 모양의 오브젝트를 만들어 보자.



## 5.2 조이스틱 버튼으로 눈사람 이동시키기

### 눈사람 이동시키는 함수

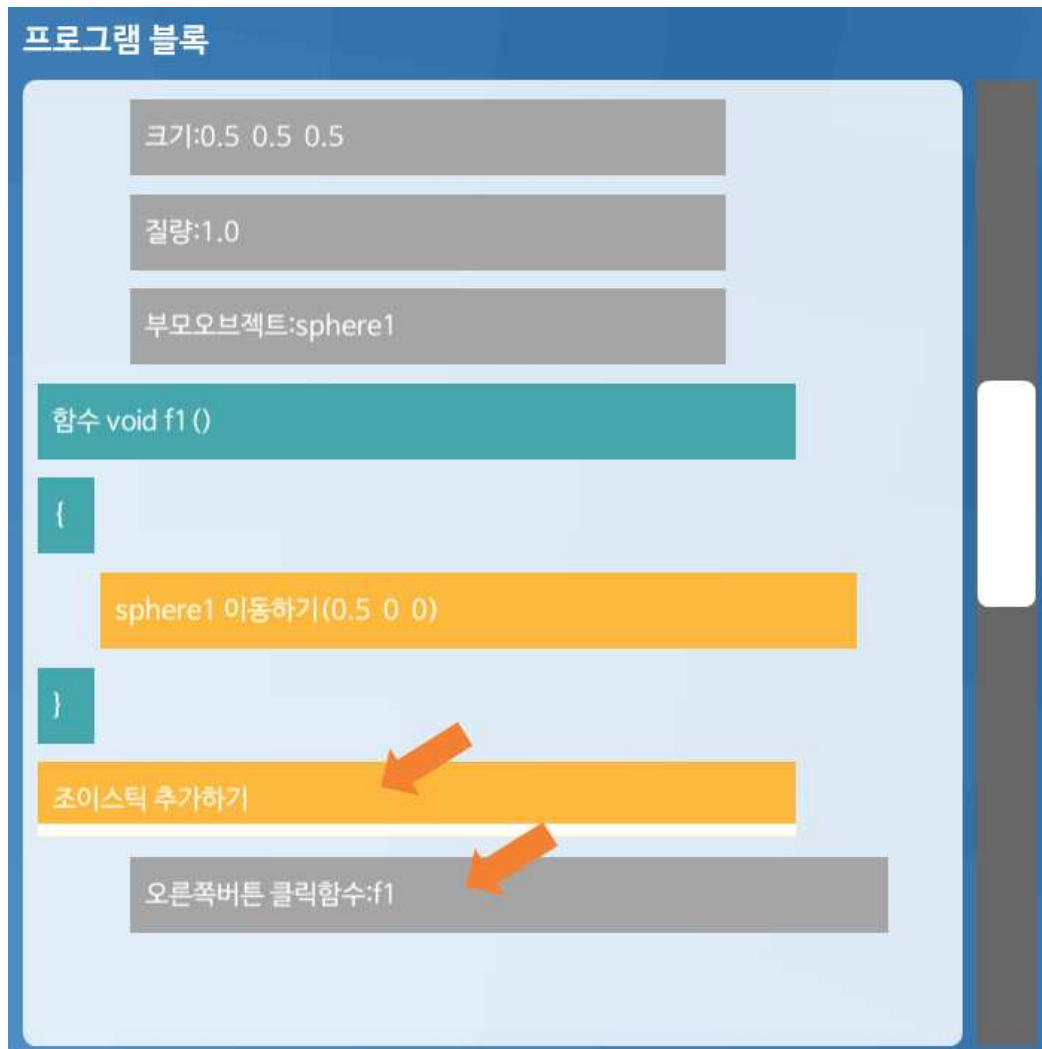
- 조이스틱의 오른쪽 버튼이 눌러질 때 마다 눈사람 오브젝트를 옆으로 이동시키려고 한다. 조이스틱 버튼이 눌러질 때 마다 호출될 수 있는 함수를 추가한 후 도구 명령어에서 이동하기 명령어를 함수 안에 추가해 준다.

The image shows a programming environment with two main components. On the left is a '프로그램 블록' (Program Block) editor. It contains several property blocks for a sphere object: '질량:1.0', '공모양 추가하기 sphere2', '위치:0 0.75 0', '크기:0.5 0.5 0.5', '질량:1.0', and '부모오브젝트:sphere1'. Below these is a function definition block: '함수 void f1 ()' followed by a code block containing 'sphere1 이동하기(0.5 0 0)'. Red arrows point from the function name and the movement command to the right-hand control panel. The control panel, titled '이동하기' (Move), has three rows for X, Y, and Z axes. Each row has a minus sign, a numeric value (0.5 for X, 0 for Y and Z), and a plus sign. Below this is a '객체 이름' (Object Name) field containing 'sphere1'. A red arrow points from the object name field to the movement command in the function block.

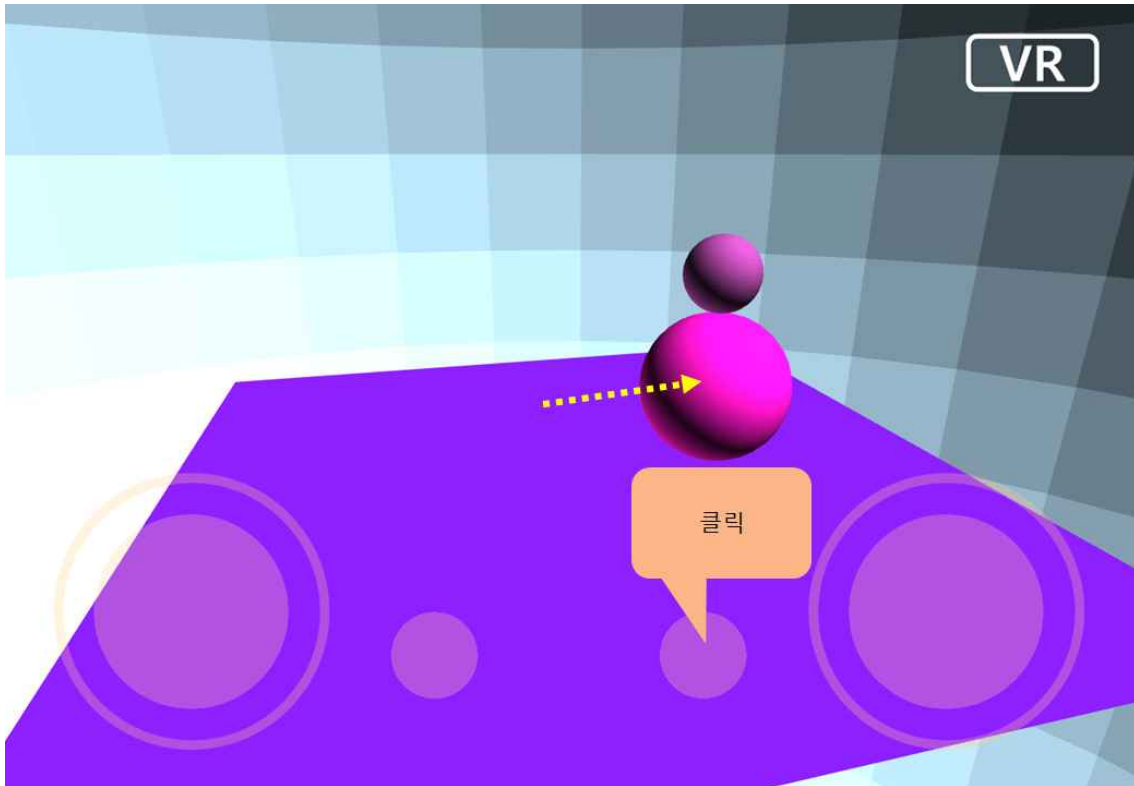
- 부모 자식으로 연결된 오브젝트를 이동시키기 위해서는 부모 오브젝트를 이동시키기만 하면 된다. 자식 오브젝트는 부모에 붙어 있기 때문에 부모가 움직이면 자동으로 같이 움직이게 된다.

## 조이스틱 버튼클릭 함수

- 조이스틱의 오른쪽 버튼이 클릭되면 함수 f1()이 호출되도록 프로그램의 맨 아래에 조이스틱 추가하기와 옵션을 추가해 준다.



- 프로그램을 실행한 후, 오른쪽 버튼을 클릭하여 눈사람이 이동하는 지 확인해 본다.



실습

- ▶ 화면에 보이는 4개의 조이스틱 버튼을 활용하여 눈사람을 여러 방향으로 이동시키는 기능을 완성해 본다.
- ▶ 눈사람 오브젝트에 힘을 가하여 넘어뜨리려고 한다. 필요한 기능을 함수에 추가하여 기능을 완성해 보자.

## 5.3 오브젝트 회전시키기

### 물체 회전시키기

- 박스 모양 오브젝트를 추가한 후, 함수를 추가하여 추가된 박스를 회전시키는 명령어를 함수 내부에 추가시킨다.

The image shows a code editor on the left and a control panel on the right. The code editor contains the following code:

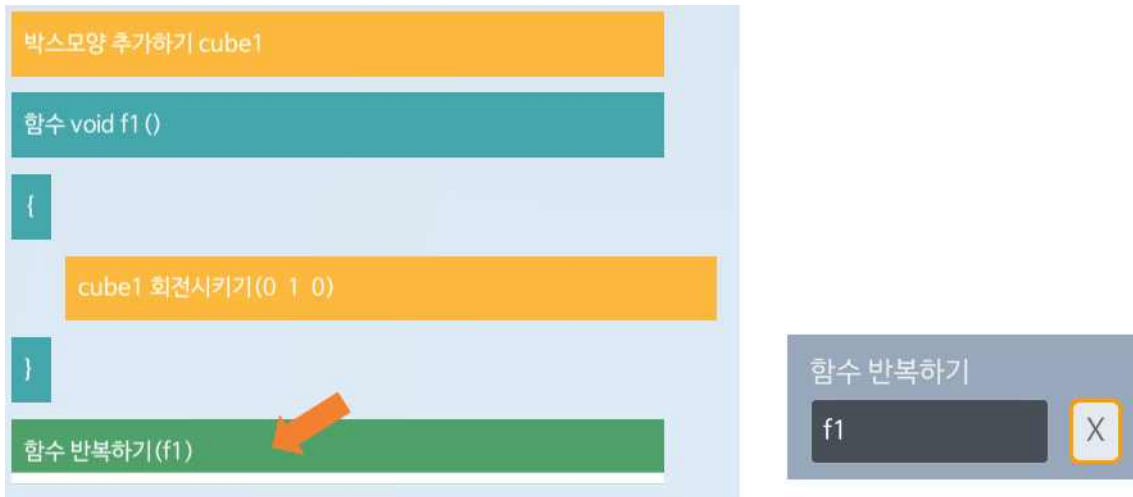
```
박스모양 추가하기 cube1  
  
함수 void f1 ()  
{  
    cube1 회전시키기(0 1 0)  
}
```

An orange arrow points to the `cube1 회전시키기(0 1 0)` line. The control panel on the right is titled "회전시키기" and has three rows of buttons for X, Y, and Z axes. Each row has a minus sign, a number, and a plus sign. The X row has 0, the Y row has 1, and the Z row has 0. Below the buttons is a section titled "객체 이름" with a text input field containing "cube1".

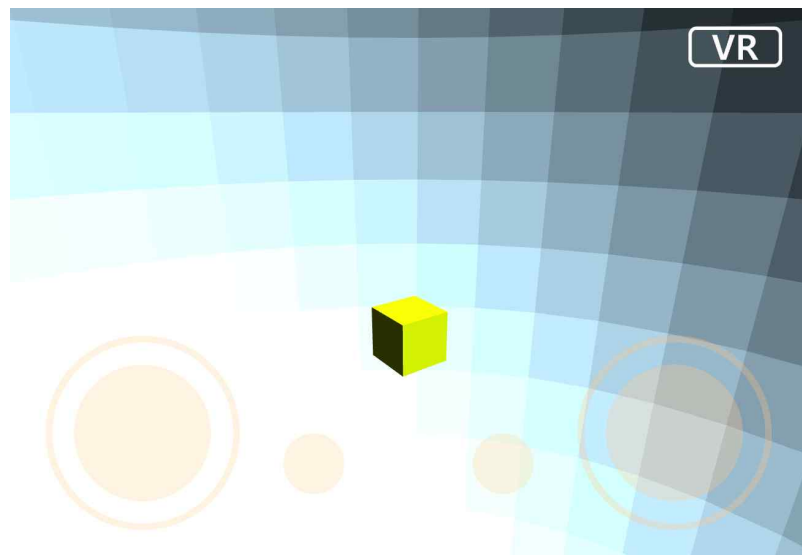
- 위의 코드를 실행하면 박스 모양 상자가 회전하지 않는다. 아직 함수를 호출해 주는 기능이 추가되어 있지 않기 때문이다.

### 함수 반복하기

- 로직 명령어 그룹의 맨 아래에는 함수 반복하기 명령어가 있다. 이 명령어를 추가해 보자.



- 프로그램을 실행하면 박스 모양이 계속 회전하는 것을 볼 수 있다.



### 실습

- ▶ 박스의 회전 방향이 반대가 되도록 하려고 한다. 위의 코드에서 어느 부분을 수정해야 할지 생각해 보고 코드를 수정해 본다.
- ▶ 박스의 회전 속도를 더 증가시켜 본다.

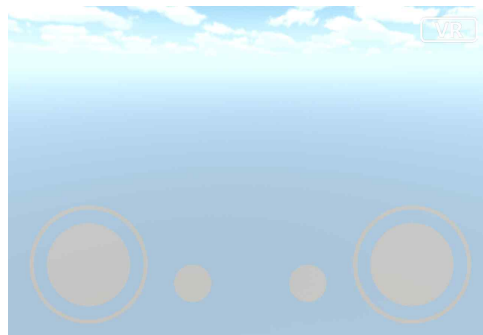
## 5.4 하늘 설정하기

### 하늘 설정하기

- 지금까지의 예제는 가상현실 코딩의 배경이 구체 형태의 모양이었다.
- 하늘 설정하기 명령어를 사용하면, 구름이 있는 하늘이나 우주와 같은 배경 그림을 적용시킬 수 있다.
- 환경 명령어 그룹에서 하늘 설정하기 명령어를 추가한 후, 옵션에서 하늘 종류 옵션을 추가한다. 기본 값은 Sunny2Skybox이다.

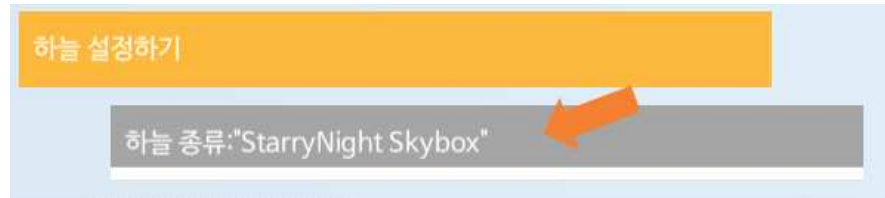


- 프로그램을 실행하여 결과를 확인해 본다.



## 어두운 밤하늘 적용하기

- 하늘 종류를 아래와 같이 어두운 밤하늘로 변경해 보자.



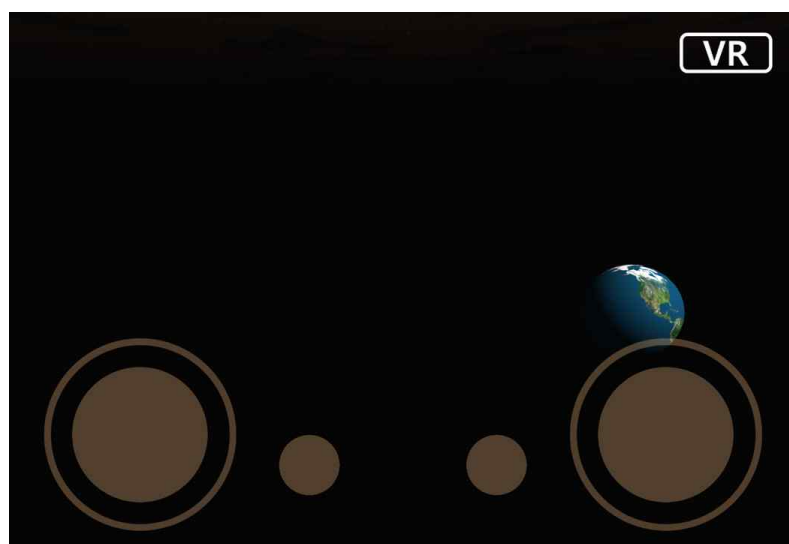
- 프로그램을 실행하면 배경이 어두운 하늘로 변경되는 것을 볼 수 있다.



## 5.5 지구 추가하기

### 지구 추가하기

- 공모양 추가하기 명령어를 추가한 후 위치와 크기, 재질을 다음과 같이 설정해 준다.





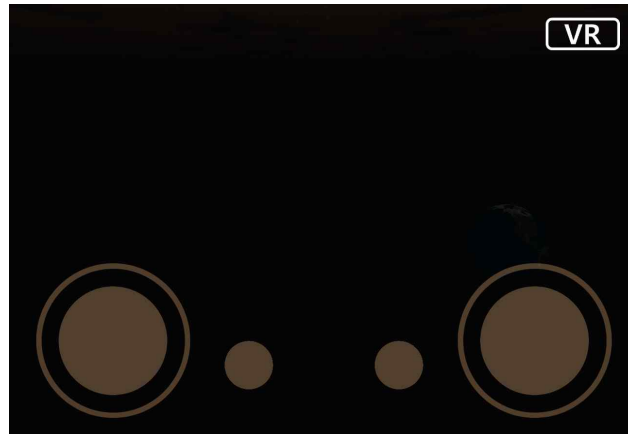
- 기본 광원이 적용되어 있기 때문에 지구가 부분적으로 밝아 보인다.

## 밝기 조절하기

- 하늘 설정하기에서는 하늘 종류외에도 공간의 전체적인 밝기를 설정할 수 있다. 기본 값은 밝은 값으로 설정이 되어 있는데, 우주 및 태양계를 만들 것이기 때문에 기본 밝기를 어둡게 설정할 필요가 있다.
- 하늘 설정하기 명령어의 옵션에서 밝기 옵션을 추가한 후, 밝기 값을 다음과 같이 어둡게 설정해 보자.



- 밝기를 어둡게 하고 다시 실행해 보면 지구가 거의 보이지 않을 정도로 어둡게 보이는 것을 알 수 있다.



- 이제 태양이 빛을 비추듯이 공간상에 광원을 추가하여 태양 효과처럼 보이도록 해보자.

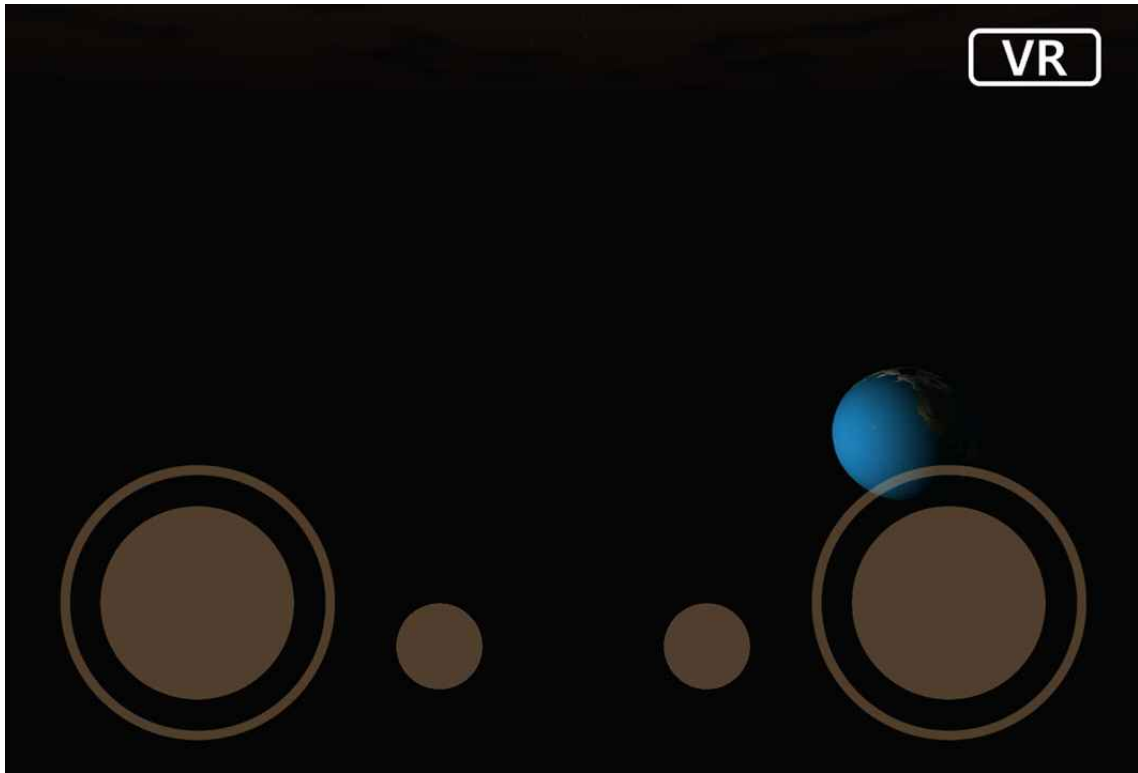
## 태양 조명 추가하기

- 환경 명령어 그룹에서 포인트라이트 추가하기 명령어를 프로그램 맨 아래에 추가한다.



- 옵션에서 빛 세기와 빛 범위 옵션을 추가한 후, 위와 같이 값을 설정해 준다.

- 프로그램을 실행하고 결과를 살펴보면 화면 중앙에서 태양 빛이 나오는 것과 같은 효과가 지구에 적용되는 것을 볼 수 있다.



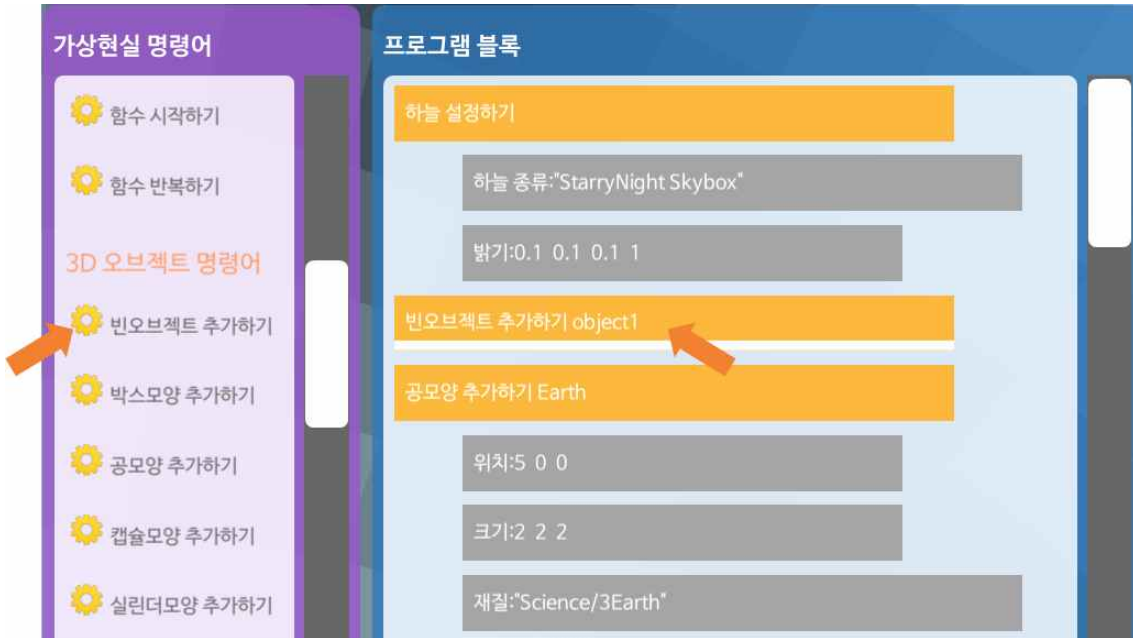
## 5.6 지구 공전 및 자전시키기

### 지구 공전축 오브젝트 추가하기

- 지구는 화면의 중앙을 기준으로 공전하도록 설정하려고 한다. 화면의 중앙에 지구 공전축 역할을 하는 빈 오브젝트를 하나 추가하고, 이 오브젝트를 지구의 부모 오브젝트로 지정한다.
- 지구는 공전축 오브젝트에 자식으로 묶여 있기 때문에 공전축 오브젝트를 공전시키면 지구가 공전하는 효과가 발생을 한다.
- 먼저 기존에 sphere1으로 되어 있던 지구 모양의 공 이름을 Earth로 수정해 준다.



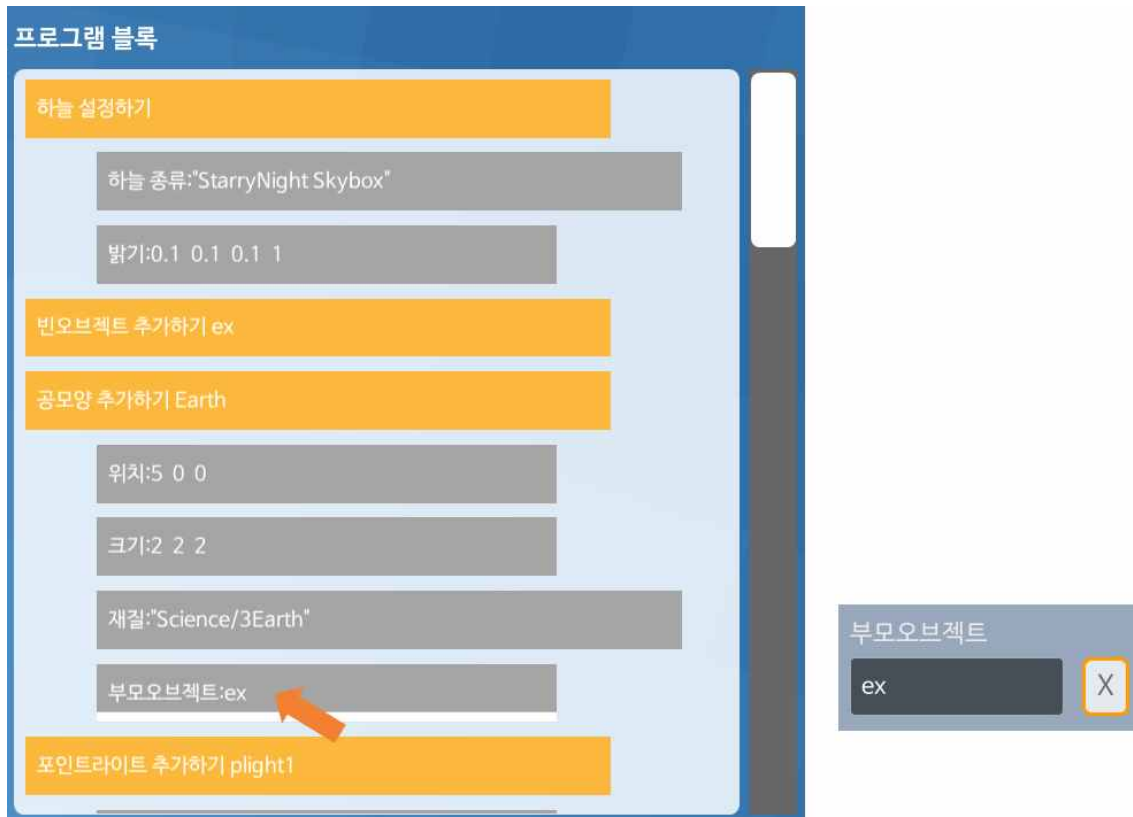
- 하늘 설정하기 명령어를 선택한 상태에서 빈오브젝트 추가하기 명령어를 클릭하면 하늘 설정하기 명령어 아래에 빈오브젝트 추가하기 명령어가 추가된다.



- 이 오브젝트의 이름을 ex로 수정해 준다.



- 지구 오브젝트에 부모오브젝트 옵션을 추가하고 부모 이름으로 ex를 지정해 준다.



### 지구 공전축 회전시키기

- 지구 공전축을 회전시키기 위해서는 지구 공전축 오브젝트를 회전시키는 함수를 하나 추가한 후, 함수 반복하기 명령어를 이용하여 이 함수를 반복적으로 호출해 주기만 하면 된다.
- 다음 코드와 같이 함수를 하나 추가한 후, ex 오브젝트를 회전시키는 명령어를 함수에 추가시킨다. 도구 명령어 그룹에서 회전시키기 명령어를 추가한 후, 객체 이름에 ex를 입력해 준다.

프로그램 블록

부모오브젝트:ex

포인트라이트 추가하기 plight1

빛 세기:3

빛 범위:70

함수 void f1 ()

```
{  
  ex 회전시키기(0 -1 0)  
}
```

회전시키기

|   |   |    |   |
|---|---|----|---|
| X | - | 0  | + |
| Y | - | -1 | + |
| Z | - | 0  | + |

객체 이름

ex

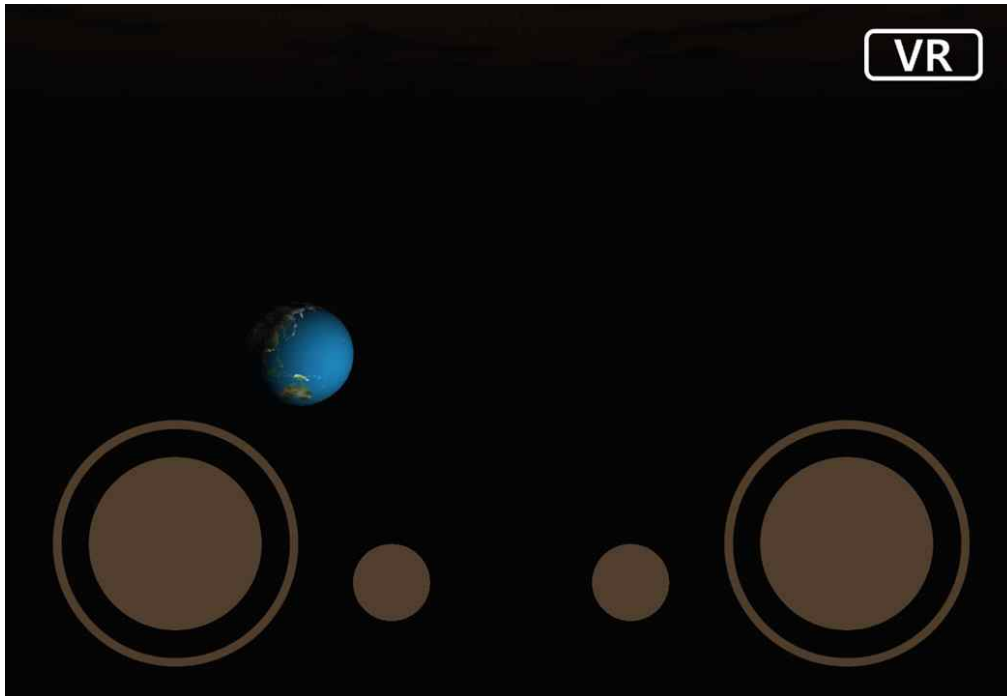
- 회전시키기 명령어에서 Y축의 값을 -1로 설정해 준다.



- 로직명령어 그룹에 있는 함수 반복하기 명령어를 프로그램 맨 아래에 추가시켜 준다.

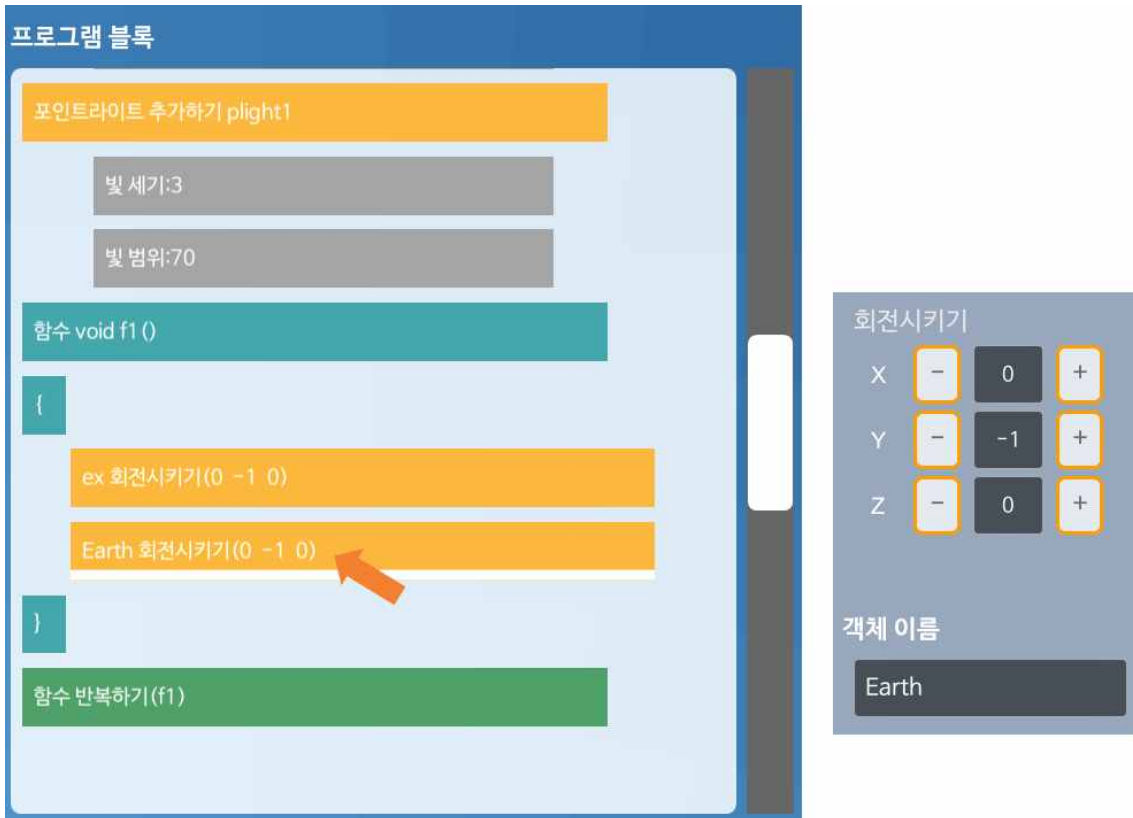


- 프로그램을 실행하면 지구가 공전하는 것을 볼 수 있을 것이다.

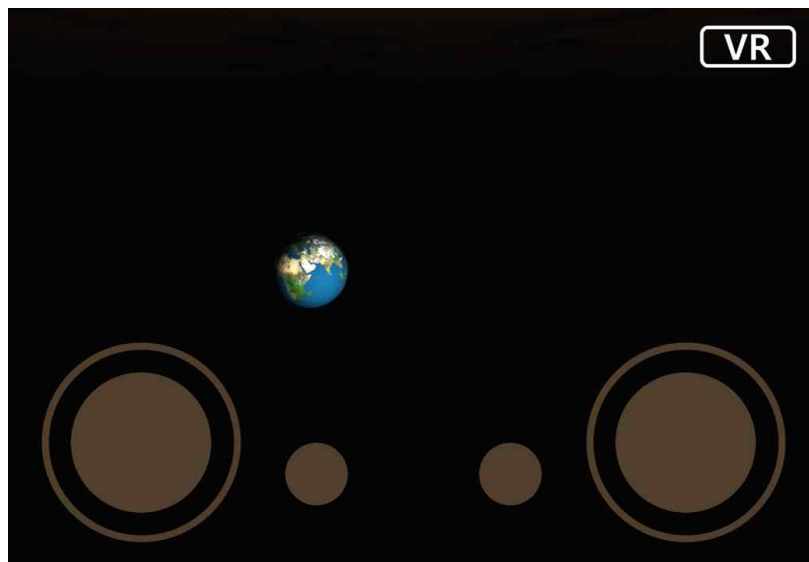


#### 지구 자전 시키기

- 지구를 자전시키는 방법은 지구 오브젝트를 직접 회전시키는 것이다.
- 함수  $f1()$  안에 다음과 같이 Earth 오브젝트를 회전시키는 명령어를 추가해 준다.



- 프로그램을 실행하면 지구가 자전함과 동시에 공전하는 결과를 관찰할 수 있다.



## 5.7 달 공전시키기

### 달 공전축 오브젝트 추가하기

- 달을 지구의 중심을 기준으로 공전하도록 설정하려고 한다. 지구와 같은 위치에 달 공전축 역할을 하는 빈 오브젝트를 하나 추가하고, 이 오브젝트를 달의 부모 오브젝트로 지정한다.
- 먼저 프로그램 맨 아래에 빈오브젝트를 추가하고 이름을 mx로 설정한다. 이 오브젝트는 달 공전축의 역할을 하며, 지구와 같이 움직여야 하므로, 이 오브젝트의 부모오브젝트는 Earth가 된다.

The image shows a programming environment with a '프로그램 블록' (Program Blocks) panel on the left and a '부모오브젝트' (Parent Object) panel on the right.

**프로그램 블록 (Program Blocks):**

- 빛 범위:70
- 함수 void f1 ()
  - {
  - ex 회전시키기(0 -1 0)
  - Earth 회전시키기(0 -1 0)
  - }
- 함수 반복하기(f1)
- 빈오브젝트 추가하기 mx
  - 부모오브젝트:Earth

**부모오브젝트 (Parent Object):**

- Earth

An orange arrow points to the '부모오브젝트:Earth' field in the '빈오브젝트 추가하기 mx' block.

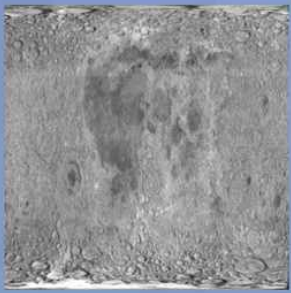
## 달 추가하기

- 프로그램 맨 아래에 공모양 추가하기 명령어를 추가한 후, 다음과 같이 설정해 준다.

**프로그램 블록**

- )
- 함수 반복하기 (f1)
- 반오브젝트 추가하기 mx
  - 부모오브젝트:Earth
- 공모양 추가하기 Moon
- 부모오브젝트:mx
- 위치:2 0 0
- 크기:0.5 0.5 0.5
- 재질:Science/3Moon2\_2500x1250\*

**옵션**



**속성값**

재질: Science/3Moon2\_2500x1250\*

사진 입히기

- 갤러리에서 사진 선택하기
- 카메라로 사진 찍기

- 프로그램을 실행하면 다음과 같이 달이 지구와 같이 공전하는 것을 볼 수 있을 것이다.

