03 슈팅 게임 만들어 보기

학습 목표

- 힘과 마찰력 등 물리 역할 실험에 필요한 기능을 활용할 수 있다.
- 힘과 무중력 현상을 이용하여 공을 발사시키고 간단한 게임으로 응용할 수 있다.

실습 개요

- 힘 가하기 명령어를 활용해 본다.
- 마찰력을 설정해 본다.
- 완전탄성 충돌을 만들어 본다.
- 힘 가하기 명령어를 이용하여 공을 발사해 본다.
- 간단한 슈팅 게임을 창작해 본다.

3.1 힘 가하기 명령어

힘 가하기 명령어

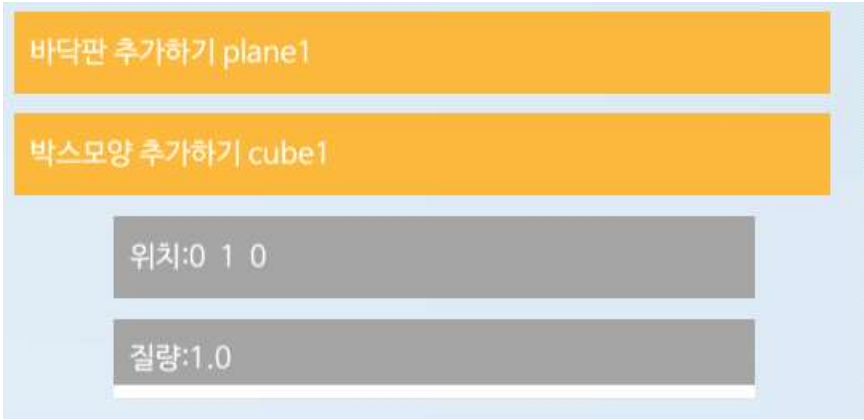
■ 도구 명령어 그룹에 포함되어 있는 힘 가하기 명령어를 사용하면 물체를 밀거나 회전시킬 수 있다. 실제 사람이 손으로 힘을 주어서 밀거나 회전시키는 것과 동일하게 작동되며, 가상으로 힘을 전달할 수 있다.

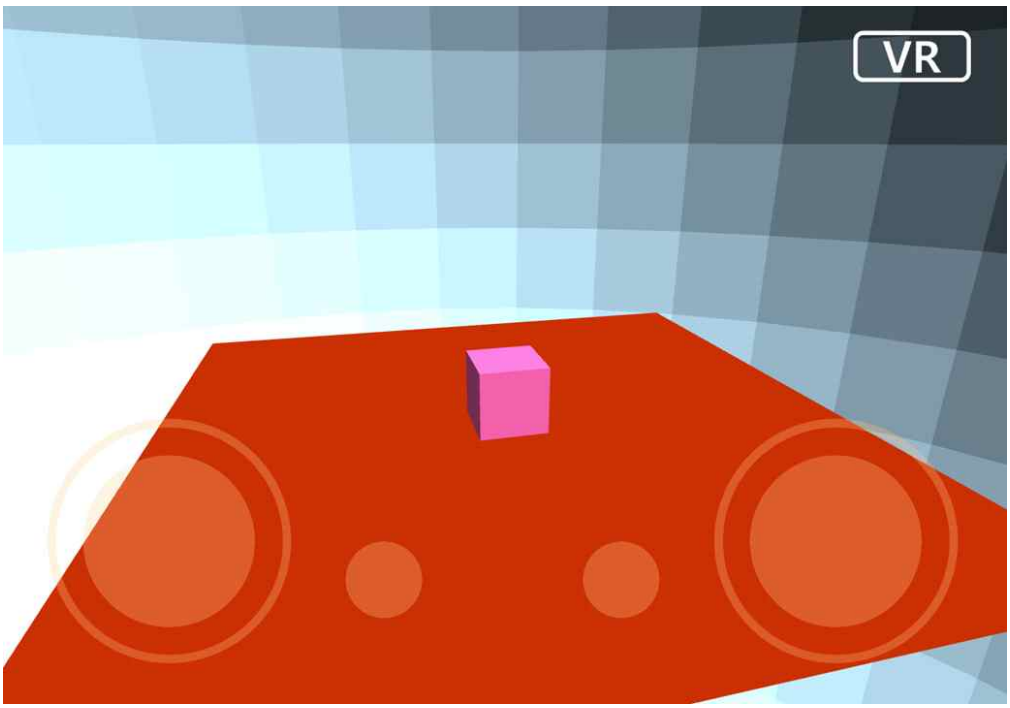
<table>
<tr>
<td> 
</td>
<td>힘 가하기:
X축, Y축, Z축의 방향에 대해 힘을 가할 수 있다. N단위를 사용한다.

회전력 가하기:
X축, Y축, Z축의 방향에 대해 토크 힘을 가할 수 있다. N단위를 사용한다.</td>
</tr>
</table>

실험 환경 구성하기

- 바닥판 위에 올려진 상자에 힘을 가해서 상자가 앞쪽으로 발사되도록 하려고 한다. 먼저, 바닥판과 박스 모양을 다음과 같이 추가한다.





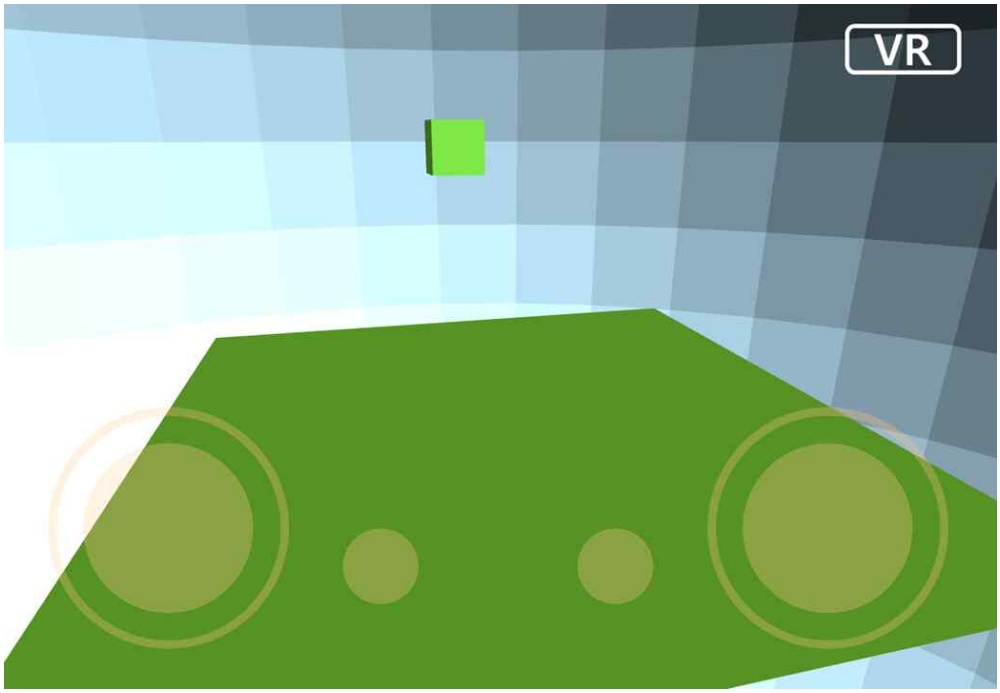
박스에 힘 가하기

- 프로그램 실행 후, 1초 후에 박스에 힘을 가하기 위해 1000 밀리초를 기다리는 명령어를 추가한다.

- 도구 명령어 그룹에 있는 힘 가하기 명령어를 추가한다.

- 힘 가하기 명령어에 있는 값의 단위는 N 단위이다.

■ 프로그램을 실행하면 1초 후에 상자가 앞으로 발사되는 것을 볼 수 있을 것이다.



실습

▶ 힘 가하기 속성값에서 힘의 값을 변경하거나 다른 축의 힘의 값을 변경하였을 때 결과가 어떻게 달라지는 지 예측해 보고 비교해 본다.

3.2 마찰력 실험하기

마찰력 설정하기

■ 3D 오브젝트에는 마찰력을 설정할 수 있다. 마찰력은 2종류가 있으며 각각 다음과 같다.

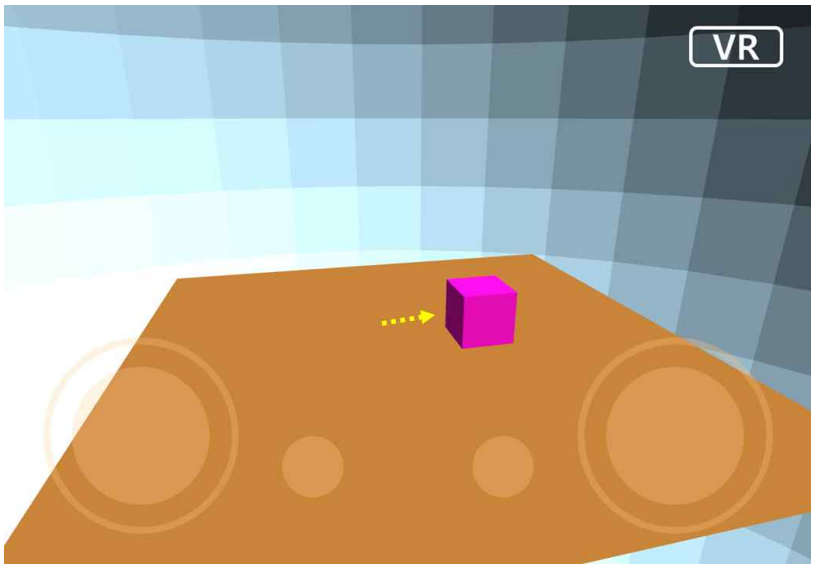
: 운동 마찰력 - 움직일 때 적용되는 마찰력 (0.0 ~ 1.0 사이의 값을 가짐)

: 정치 마찰력 - 정지해 있을 때 적용되는 마찰력 (0.0 ~ 1.0 사이의 값을 가짐)

: 마찰력 값이 0이면 마찰이 없는 것이며, 값이 1이면 마찰이 가장 큰 경우임

■ 이전 활동에서 작성한 코드에서 박스 모양에 다음과 같이 운동마찰력과 정지마찰력 옵션을 추가해 준 후, 힘 가하기 값을 아래와 같이 수정해 준다.

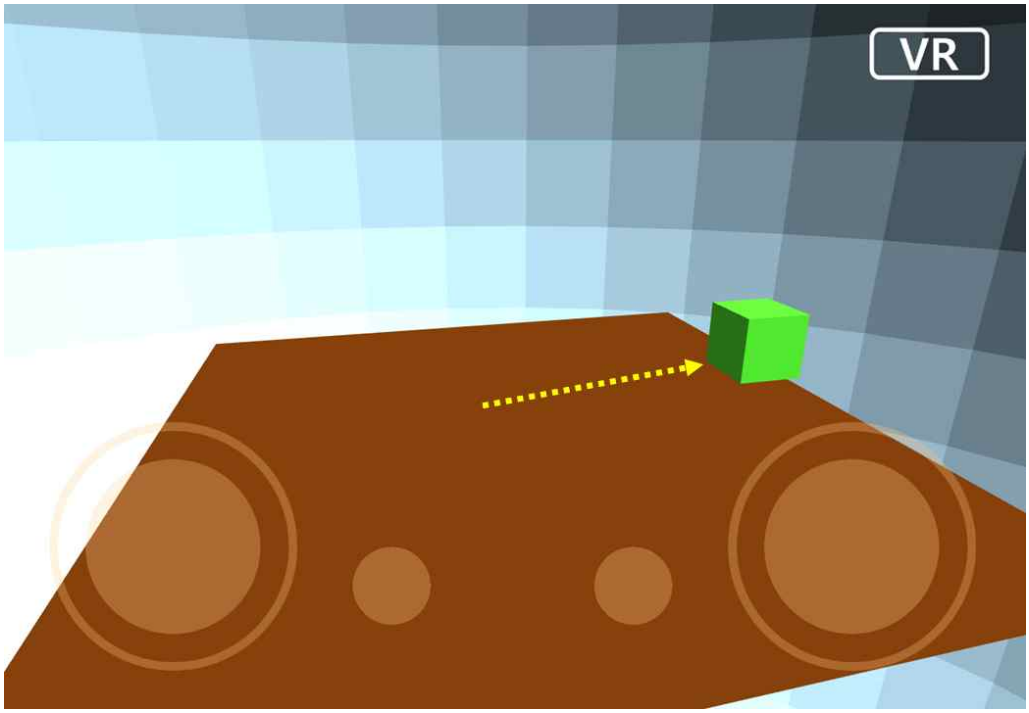
■ 코드를 실행하면 1초 후에 상자가 옆으로 이동하게 된다. 상자에는 마찰이 없지만, 아직 바닥판에는 마찰력이 적용되어 있어, 상자가 조금만 움직이고 멈추게 된다.



바닥판의 마찰력 없애기

■ 상자에 힘을 가했을 때, 상자가 계속 움직이도록 하려면 바닥판의 마찰력도 없앨 필요가 있다. 바닥판에 마찰력을 적용하려면 바닥판에 위치고정 옵션도 같이 추가되어야 한다. 기존 코드에서 바닥판 추가하기 명령어에 아래와 같이 옵션을 추가한 후, 코드를 실행시켜 본다.

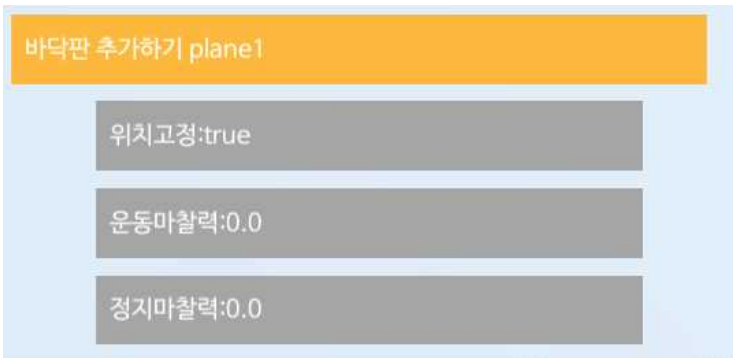
■ 코드를 실행하면, 상자가 바닥판 위를 계속 이동하게 된다.



3.3 두 개의 상자 충돌시키기

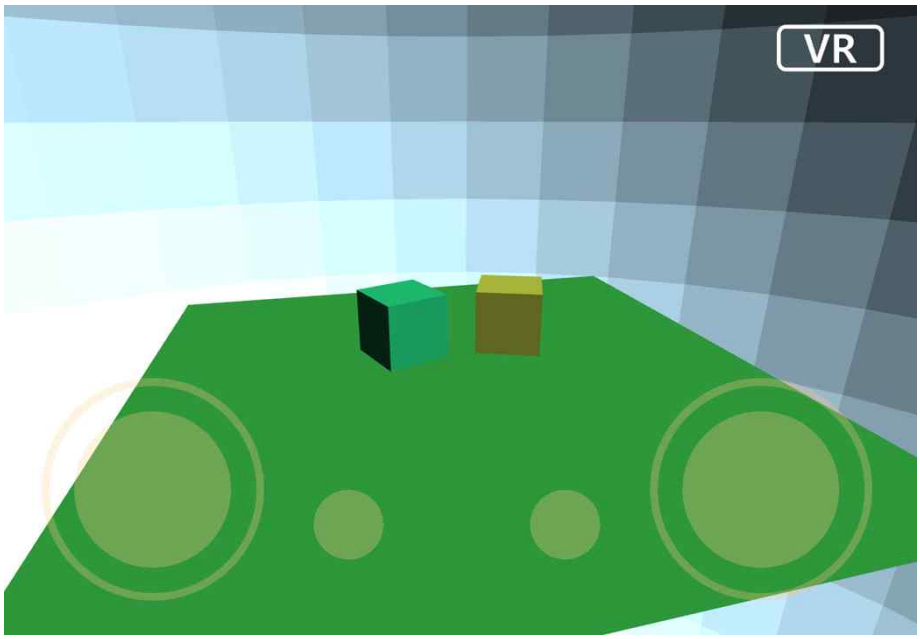
실험 환경 구성하기

- 2개의 상자를 서로 충돌시키기 위해 다음과 같이 하나의 상자를 더 추가한 후, 각각 X축으로 3m와 -3m 위치로 설정한다.



■ 기존 코드 아래 쪽에 다음과 같이 1초가 경과된 후, 각각의 상자에 서로 반대 방향으로 힘을 가해 보는 명령어를 추가해 준다.

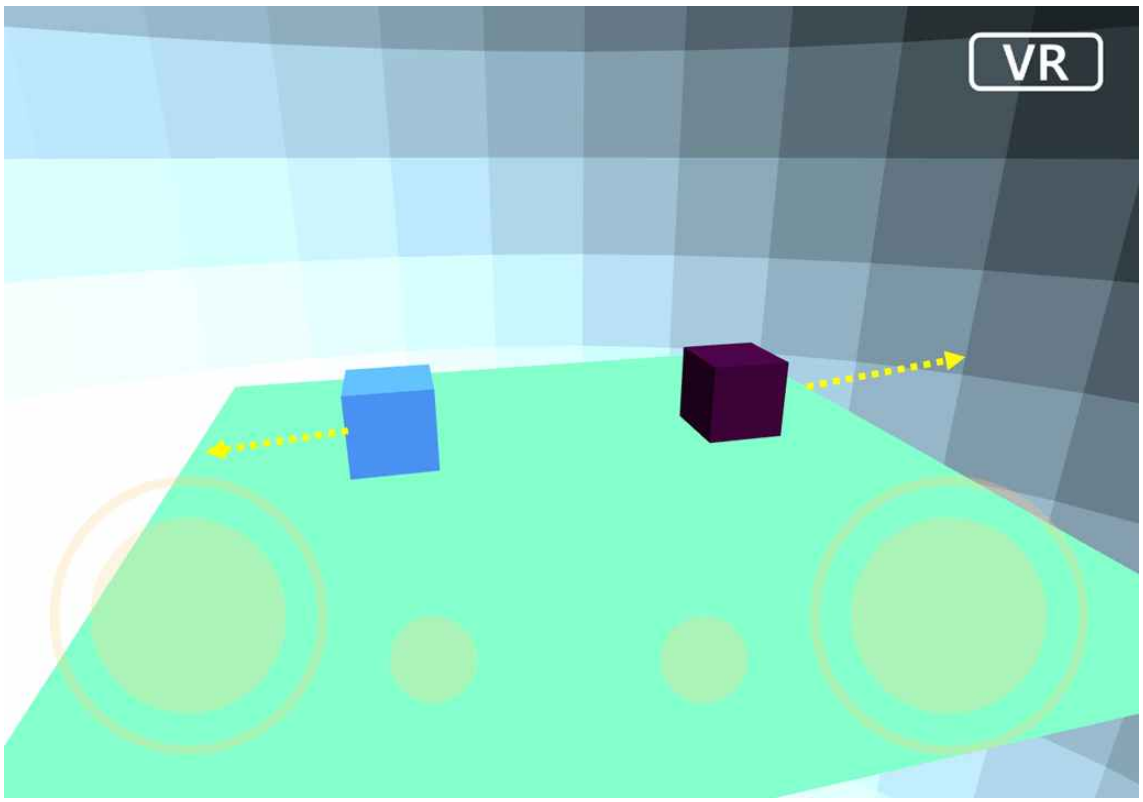
■ 코드를 실행하면 다음과 같이 1초 후에 두 개의 상자가 서로 충돌하게 되는 것을 볼 수 있다.



완전 탄성 충돌 만들기

- 2개의 상자는 서로 충돌을 하기는 하지만 상자에 탄성이 없기 때문에 상자가 서로 튕기지 않고 충돌 지점 부근에 머무르는 결과가 나오게 된다.

- 2개의 상자에 각각 탄성을 옵션을 추가하여 충돌시킬 경우 어떠한 결과가 나오게 될지 예측해 보자.



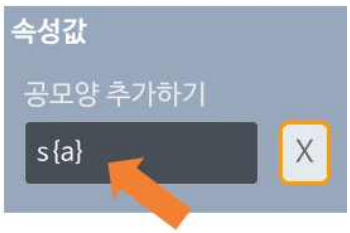
3.4 조이스틱 버튼으로 공 발사하기

공 생성 함수

- VR 공간에서 생성되는 3D 오브젝트는 각각 고유한 이름을 가져야 한다. 이름이 중복되게 되면, 오브젝트가 가상 공간상에 생성되지 않는 문제가 발생을 한다.

- 생성된 오브젝트에 힘을 가하려면 오브젝트의 이름을 반드시 알아야 하는데, 이를 위해서는 오브젝트의 이름을 변수로 처리해 주어야 한다.

- 아래의 코드는 함수에서 공 모양을 생성할 때, 함수가 호출될 때 마다 숫자를 1씩 증가시키고, 이 숫자를 공의 이름에 대입하는 사례를 보여준다. 변수 선언과 숫자를 증가시켜 주는 수식은 수식 명령어를 사용하여 입력해 준다.



- 위의 코드에서 함수 f1()이 호출될 때 마다 전역변수 a의 값은 1씩 증가하며, 공의 이름은 s1, s2, s3 … 와 같이 이름이 겹치지 않게 생성된다.

생성된 공에 힘 가하기

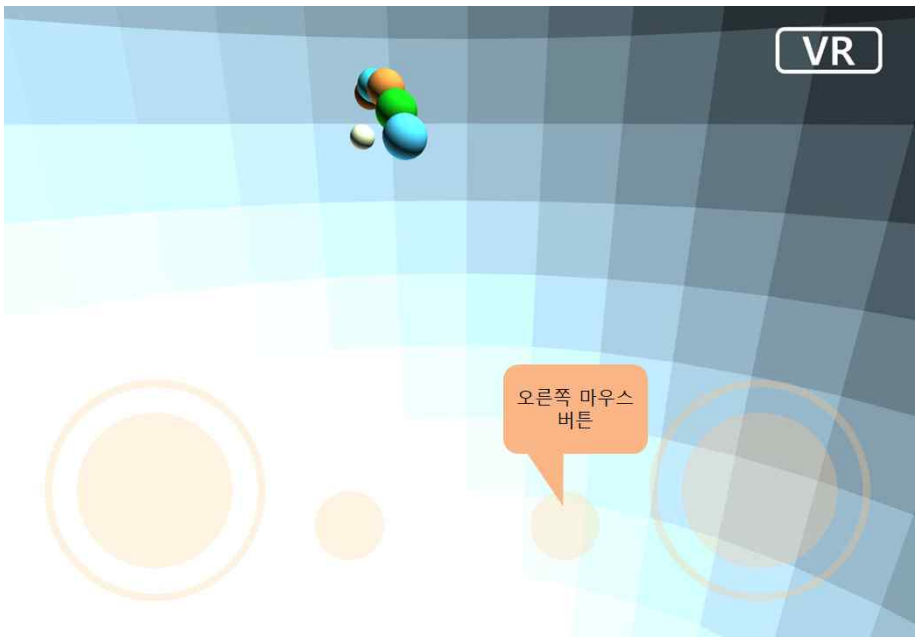
- 생성된 공에 힘을 가하기 위해서는 공의 이름을 알아야 한다. 함수에서 생성된 공의 이름은 s{a}처럼 변수로 구성이 되어 있는 데, 힘 가하기 명령어에도 동일하게 이름을 지정해 준다.

조이스틱 버튼으로 공 발사하기

- 조이스틱의 오른쪽 버튼 이벤트에 함수 f1을 호출하는 기능을 추가하면 오른쪽 버튼이 눌려졌을 때, 공이 발사되는 결과를 볼 수 있다.

- 다음과 같이 조이스틱 추가하기 명령어를 추가한 후, 옵션에서 오른쪽버튼 클릭함수 옵션을 추가해 준다.

- 코드를 실행한 후, 화면에서 오른쪽 버튼을 클릭하면 어떠한 결과가 나타나는지 관찰해 보자.



3.5 카메라 방향으로 공 발사하기

힘 가하기 함수

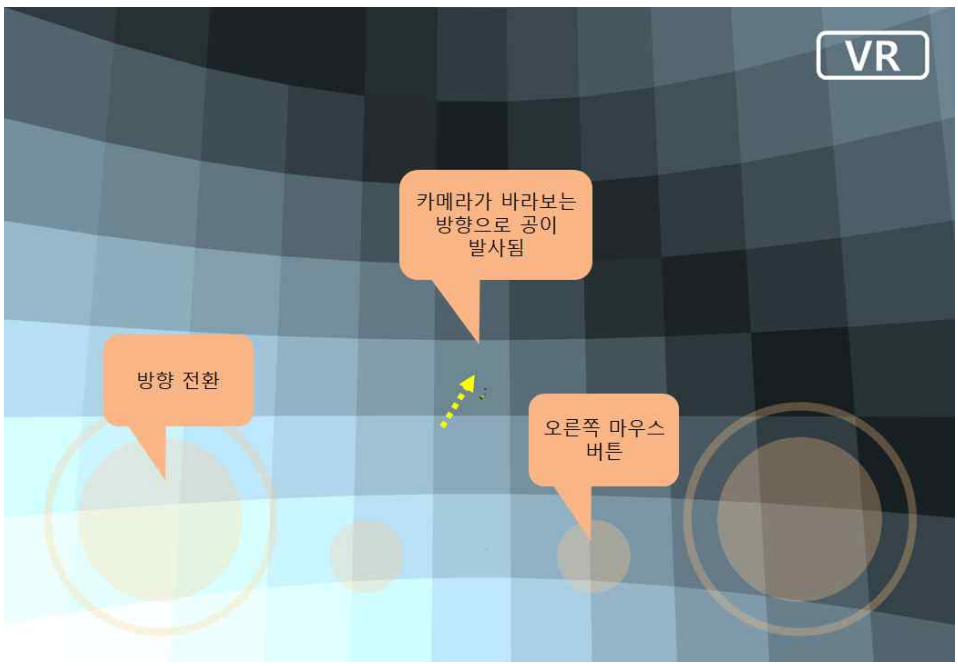
- 이전 활동의 예제를 실행하면 공이 항상 일정한 방향으로만 발사되고 공의 발사 각도를 실행중에 변경하는 것이 불가능 하였다.

- 이번에는 공의 발사 방향을 카메라가 바라보는 방향으로 항상 발사되도록 힘 가하기 값을 수정해 본다.

- 힘 가하기 명령어는 인수를 3개 또는 1개를 받을 수 있는데, 그동안 예제에서는 인수를 3개로 지정하였다. 하지만 인수를 1개만 지정할 경우에는 힘이 카메라가 바라보는 방향으로 적용되게 된다.

힘 가하기 (0, 500, 500)	Y축으로 500N, Z축으로 500N의 힘이 가해진다.
힘 가하기 (5000)	카메라가 바라보는 방향으로 5000N의 힘이 가해진다.

- 함수 f1()에 있는 힘 가하기 명령어를 다음과 같이 수정한 후 다시, 코드를 실행시켜서, 결과가 어떻게 달라지는 지 관찰해 보자.



3.6 슈팅 게임 만들기

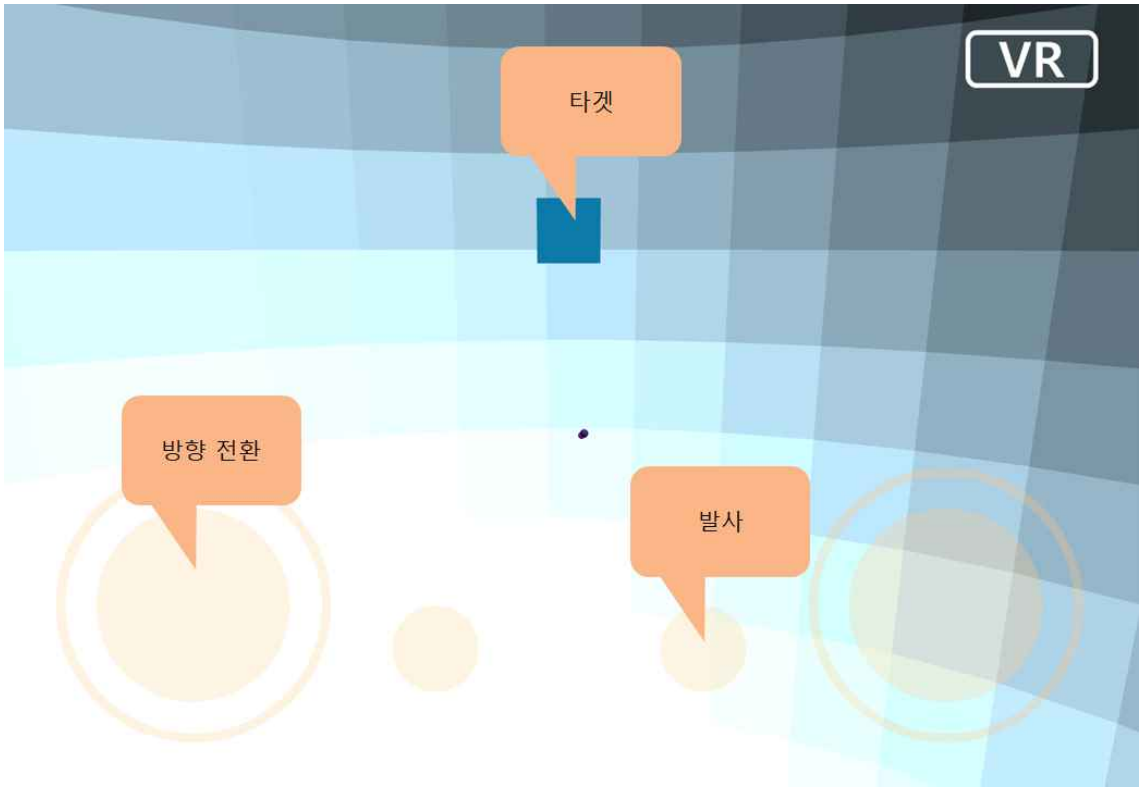
무중력 장애물 만들기

- 카메라가 바라보는 방향으로 공을 발사하게 되었으므로, 이제 공간 상에 장애물들을 배치해 놓게 되면 간단한 슈팅게임이 된다.

- 장애물은 떠 있으면서 충돌 후, 튕겨져 나가야 하므로 무중력 상태의 박스 도형으로 생성을 한다.

- 아래의 코드는 전방 위쪽 공간에 하나의 박스를 무중력 상태로 만드는 예를 보여 준다.

■ 코드를 실행하여 결과를 확인해 본다.



실습

▶ 무중력으로 떠 있는 장애물을 원하는 위치에 더 추가해 본다.

▶ 반복문과 랜덤 함수를 이용하여 많은 수의 장애물을 자동으로 생성시켜 보자.