

08 충돌 감지 및 거리측정 기능 활용하기

학습 목표

- 충돌감지 기능을 활용하여 다양한 기능을 구현할 수 있다.
- 물체들 간의 거리 측정 방법을 이해하고 응용할 수 있다.

실습 개요

- 충돌이 감지되면 메시지를 출력해 본다.
- 충돌이 감지되면 새로운 오브젝트를 생성시켜 본다.
- 거리를 측정하여 떠 있는 오브젝트를 구현해 본다.

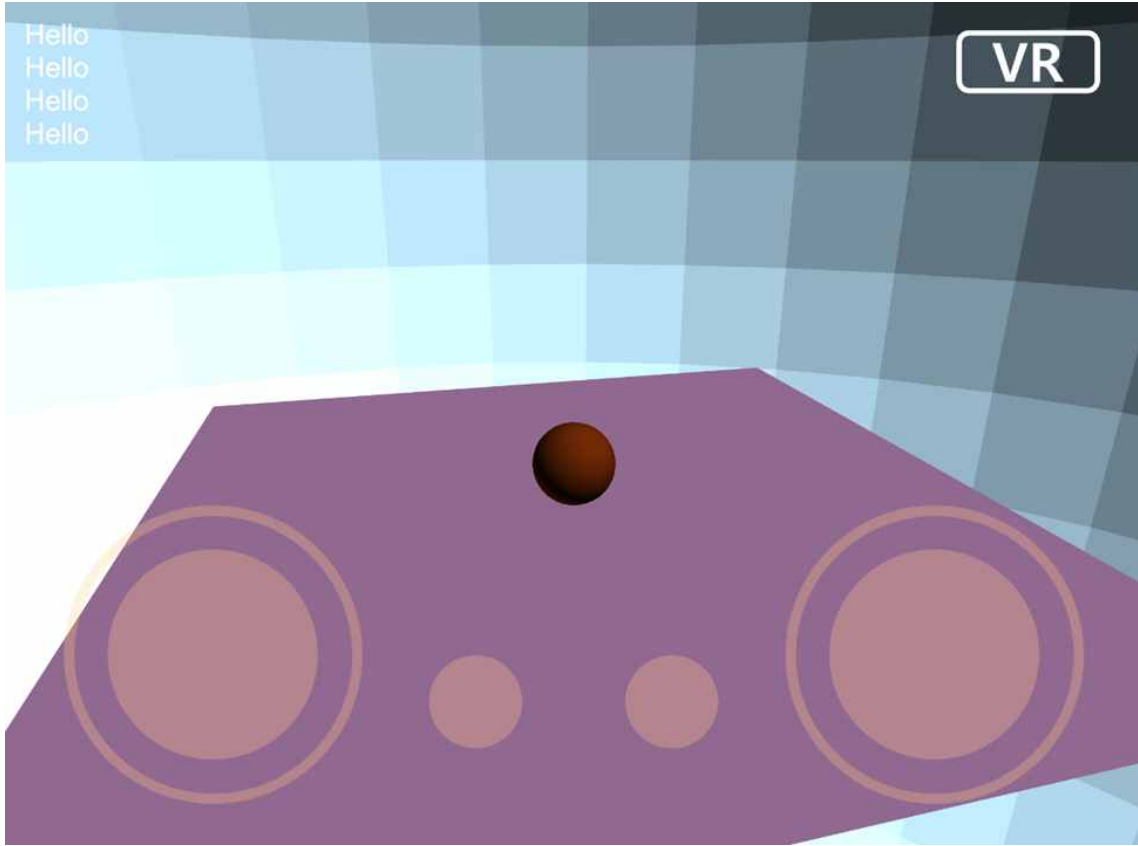
8.1 충돌 감지하기

충돌 감지 함수

- 3D 가상공간 상의 모든 오브젝트는 충돌을 감지하는 기능을 지원한다.
- 아래의 코드는 공이 바닥판으로 떨어지면서 충돌할 때 마다 메시지를 출력하는 기능을 수행한다.

```
바닥판 추가하기 plane1
공모양 추가하기 sphere1
위치:0 5 0
탄성:1.0
충돌감지 함수:f1
함수 void f1()
{
    문자라인 출력("Hello")
}
```

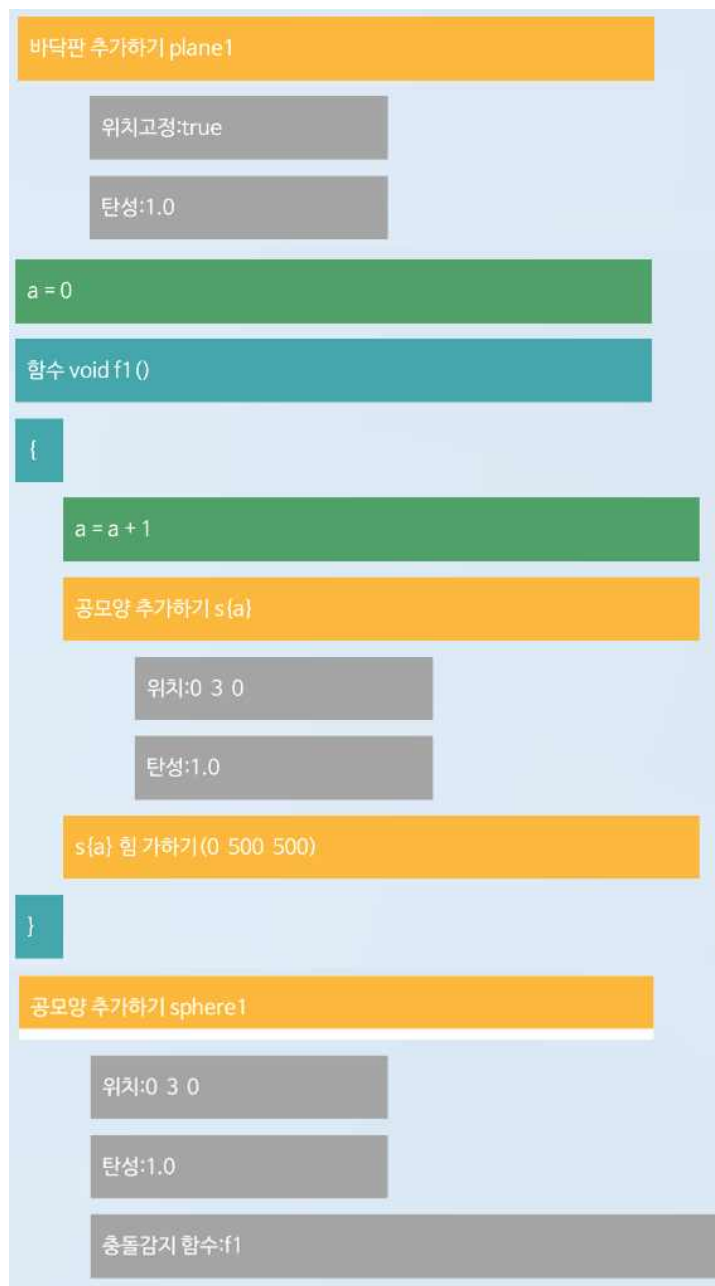
- 실행 결과는 다음과 같다.



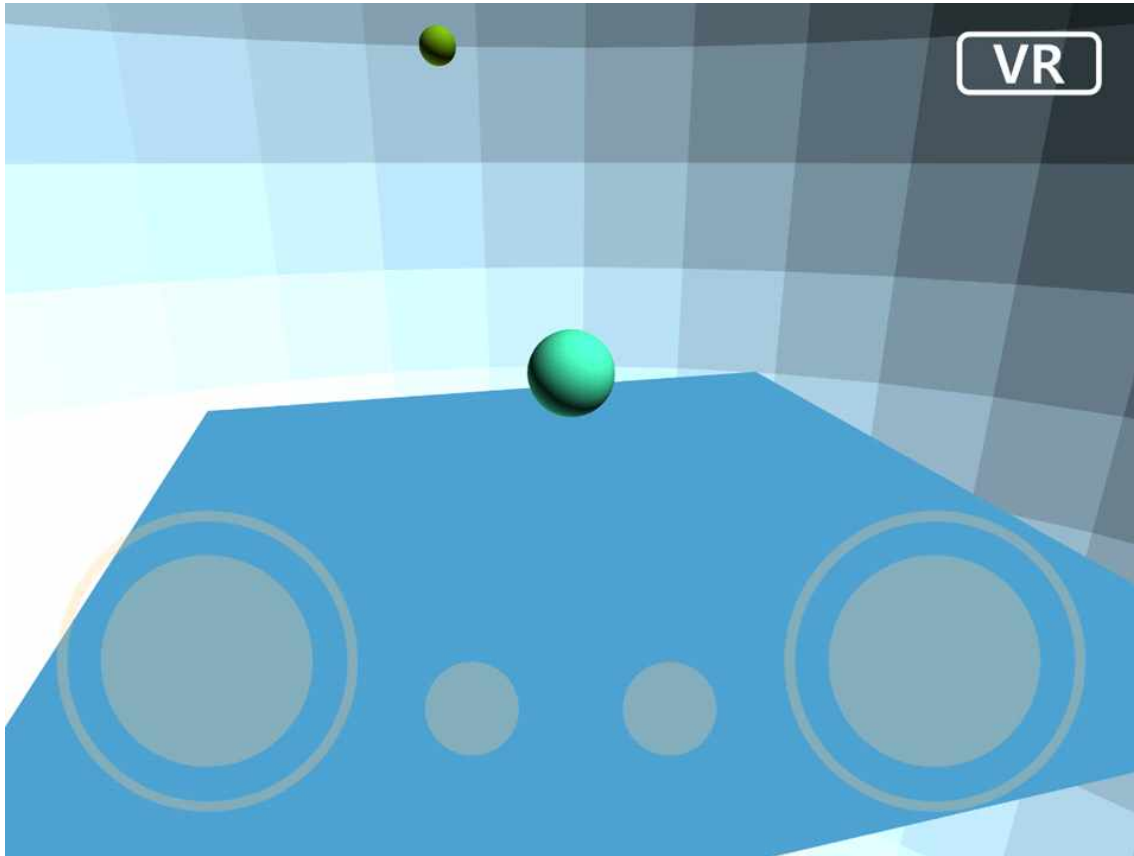
8.2 충돌할 때 마다 새로운 공 생성시키기

충돌시 공 생성하기

- 아래의 코드는 떨어지는 공이 충돌할 때 마다 새로운 공이 생성되도록 기능을 구현한 예이다.



- 실행 결과는 다음과 같다.



실습

- ▶ 충돌 이벤트를 활용할 수 있는 창의적 시나리오를 기획해 본다.
- ▶ 기획한 시나리오를 직접 구현한 후, 발표해 본다.

8.3 충돌 정보 가져오기

충돌 정보

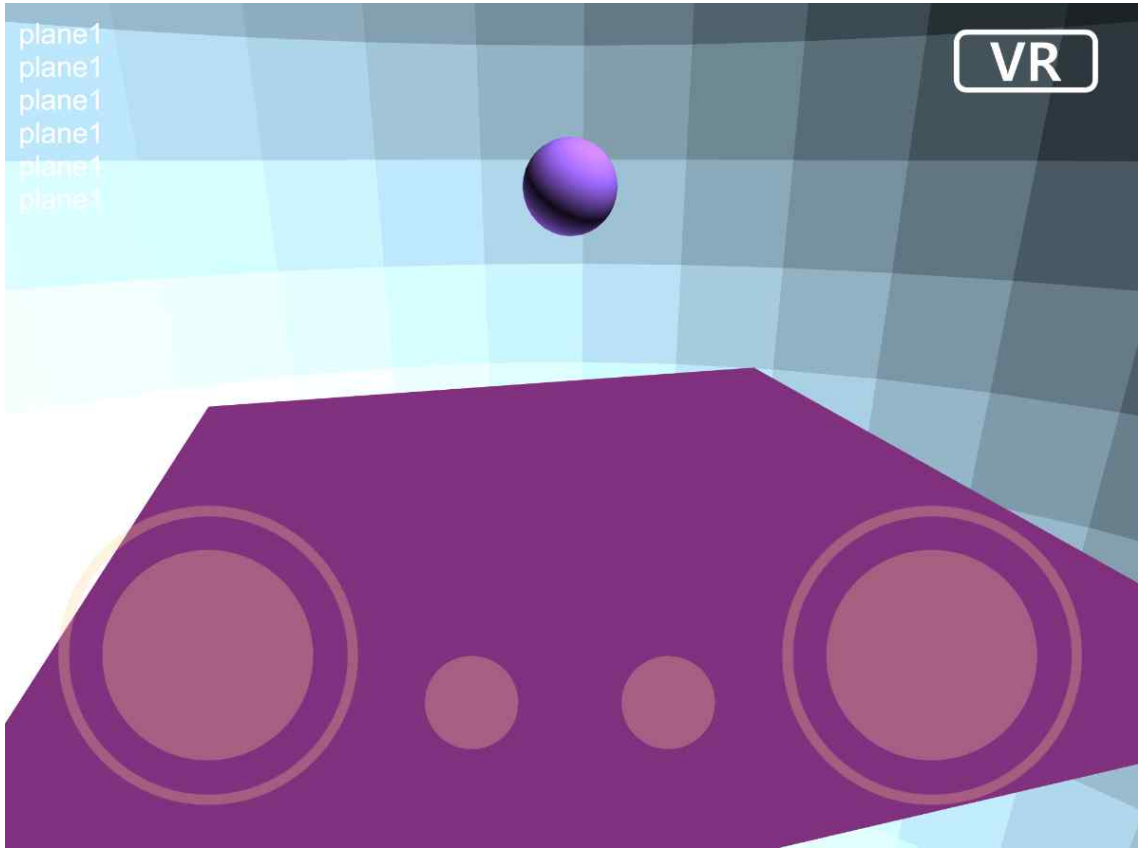
- 충돌이 감지되고 함수가 호출되면, 해당 함수의 다음의 변수 값들이 전달된다.
 - ▶ name: 현재 물체의 이름
 - ▶ target: 충돌한 물체의 이름
 - ▶ point: 충돌한 물체 상에서의 충돌 위치
- 다음의 코드는 충돌시 충돌한 물체의 이름을 출력하는 예제이다.

```
바닥판 추가하기 plane1
  위치고정:true
  탄성:1.0

함수 void f1 ()
{
  문자라인 출력(target)
}

공모양 추가하기 sphere1
  위치:0 3 0
  탄성:1.0
  충돌감지 함수:f1
```

- 실행 결과는 다음과 같다.



8.4 거리 측정 정보 가져오기

거리 측정 함수

- 3D 오브젝트에는 전방 방향으로의 다른 물체와의 거리를 측정하는 이벤트 함수가 지원된다. 최대 측정 거리는 1000m이다.

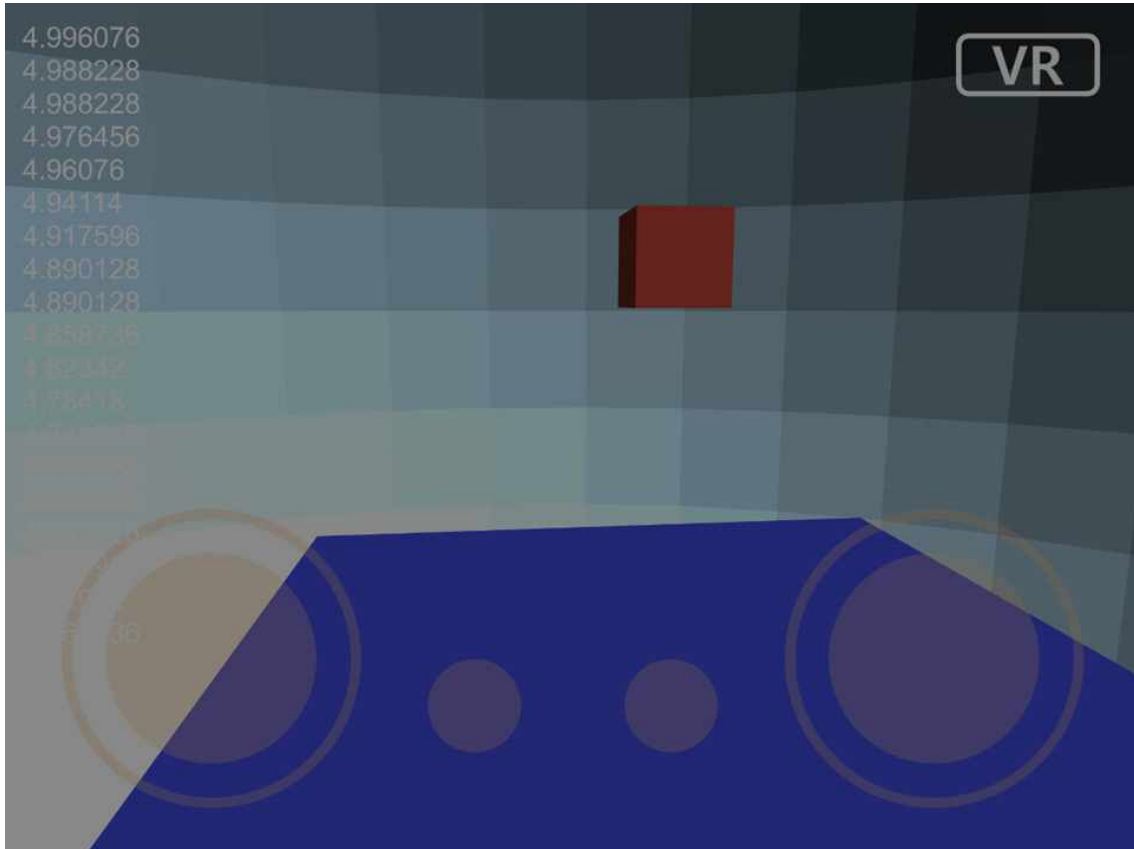
- 호출되는 함수에는 필요한 정보들이 인수로 전달되는데, 해당 값은 다음과 같다.
 - ▶ name: 현재 프로시저를 호출한 물체의 이름
 - ▶ target: 가상 빛이 닿은 물체의 이름
 - ▶ distance: 대상 물체까지의 거리
 - ▶ color: 대상 물체의 색상
 - ▶ point: 대상 물체에 빛이 닿은 위치

거리값 출력하기

- 아래의 예는 박스가 아래로 떨어지면서 아래 방향으로의 거리정보를 출력하는 예제를 보여준다.

```
바닥판 추가하기 plane1
박스모양 추가하기 cube1
  위치:0 5 0
  질량:1.0
  방향:90 0 0
  거리측정 함수:f1
함수 void f1 ()
{
  문자라인 출력(distance)
}
```

- 실행 결과는 다음과 같다.

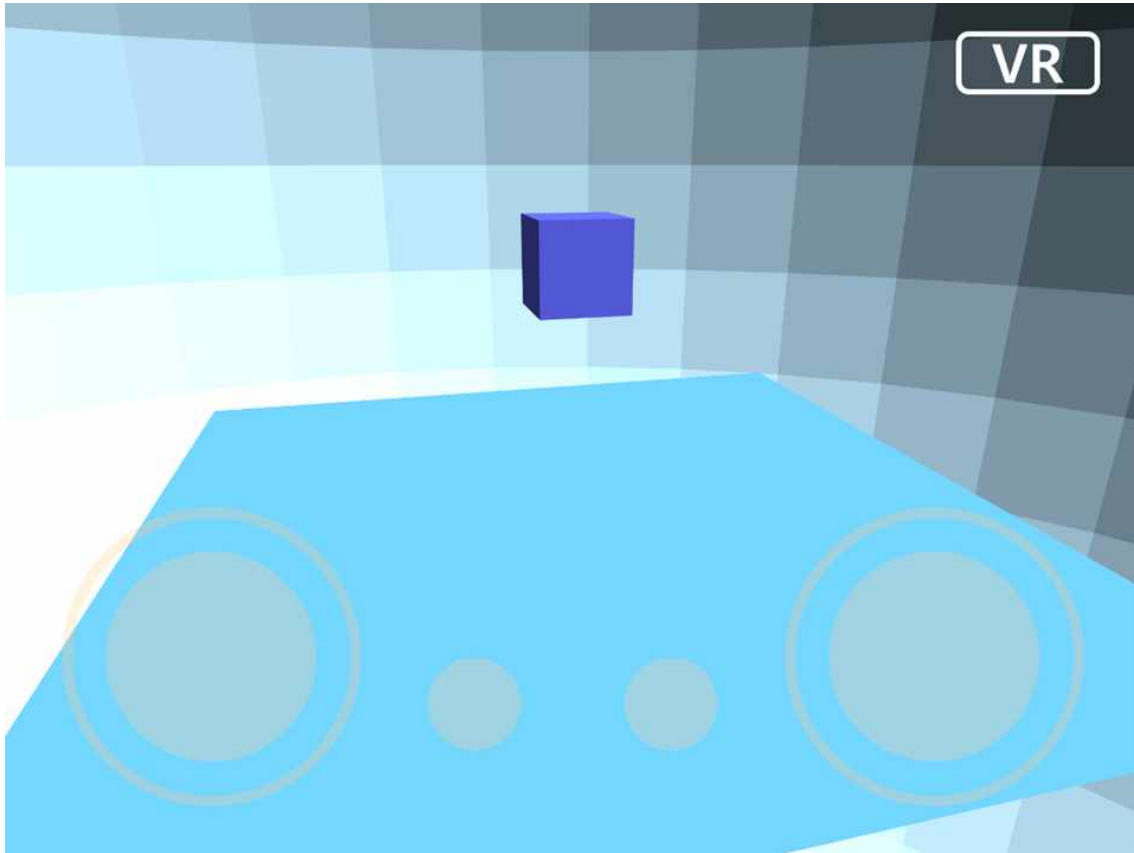


물체를 바닥에 닿지 않게 하기

- 아래의 예는 떨어지는 박스와 바닥간의 거리 정보를 이용하여 일정 거리 이내이면 다시 박스에 힘을 가해 위로 올리는 기능이 포함된 사례이다.

```
바닥판 추가하기 plane1
박스모양 추가하기 cube1
  위치:0 5 0
  질량:1.0
  방향:90 0 0
  거리측정 함수:f1
함수 void f1()
{
  if (distance < 3)
  {
    cube1 힘가하기(0 50 0)
  }
}
```

- 실행 결과는 다음과 같다.



실습

- ▶ 거리 정보를 활용하는 다양한 창작 작품 시나리오를 설계하고 구현해 본다.