

## 09 로봇카 및 자동차 제어하기

### 학습 목표

- 도로 모델을 활용할 수 있다.
- 로봇카 및 자동차 모델을 다룰 수 있다.

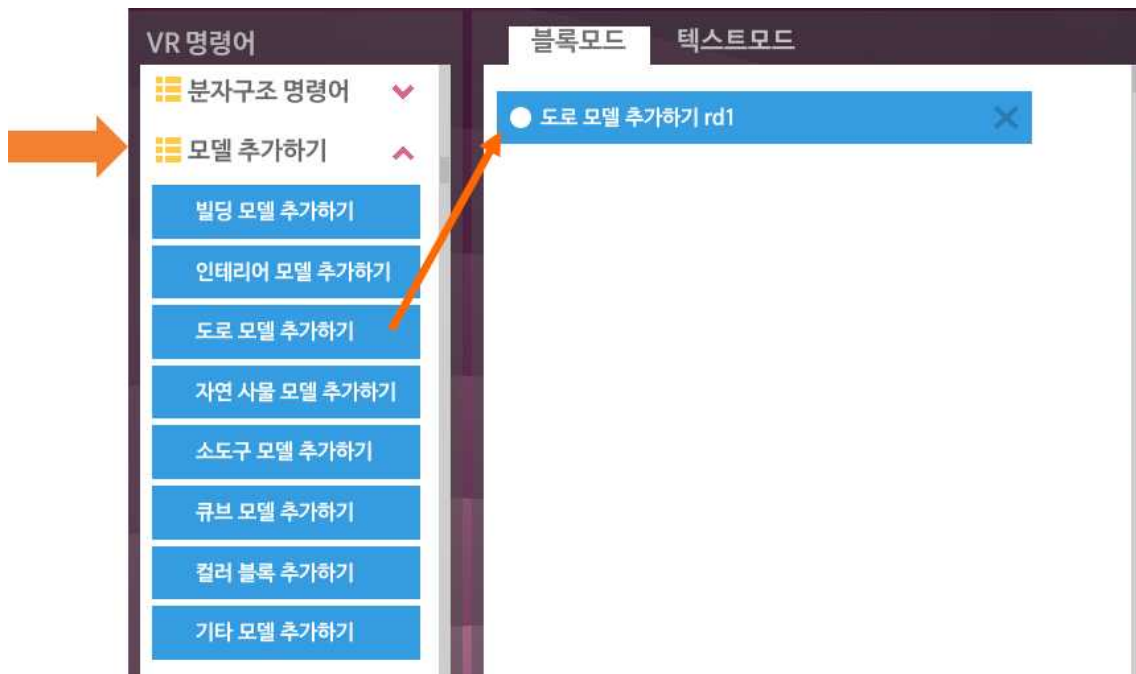
### 실습 개요

- 도로 모델링을 통해 도로를 구현해 본다.
- 로봇카를 제어해 본다.
- 로봇카에 센서를 장착하여 제어해 본다.
- 자동차를 제어해 본다.

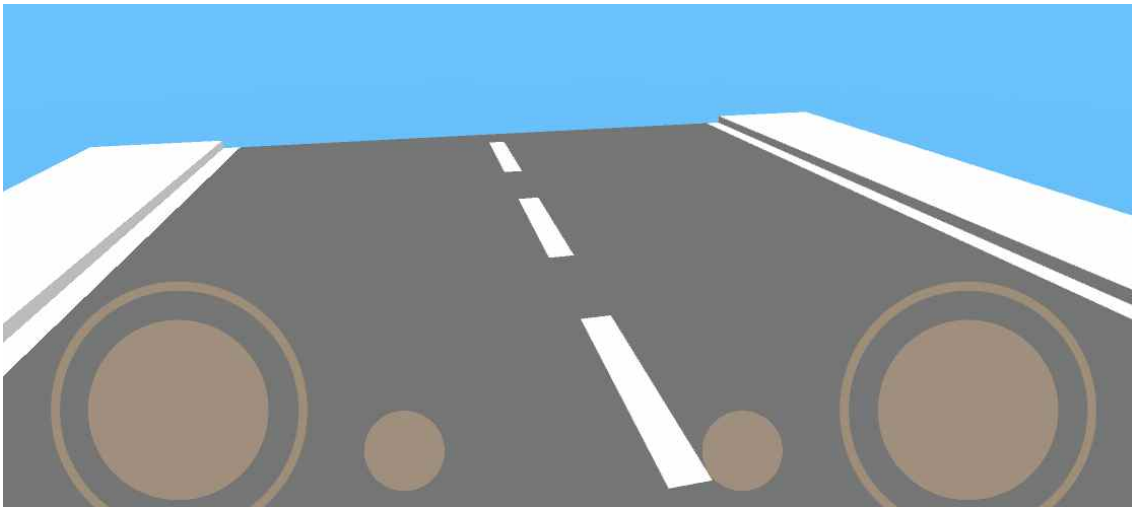
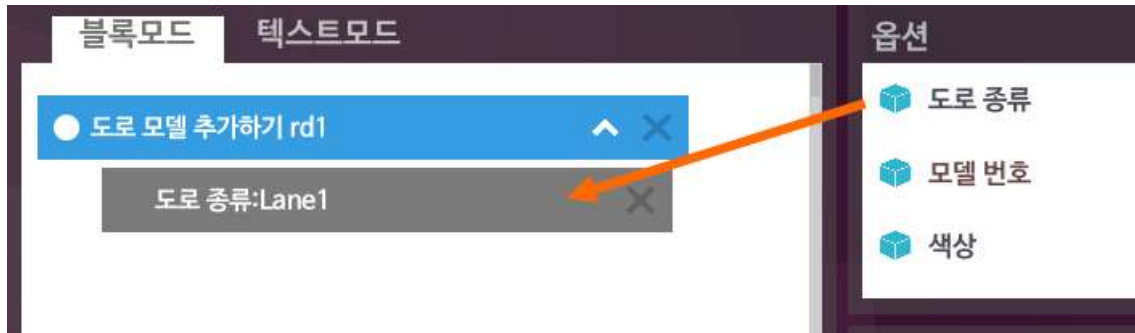
## 9.1 도로 만들기

### 도로 모델링

- 모델 추가하기 그룹에 있는 도로 모델 추가하기 명령어를 추가해 본다.



- 도로 종류 옵션을 추가한다.



- 도로 모델링이 로봇카 또는 자동차를 지탱하기 위해 충돌체 기능 가지기 옵션을 추가한다. 아래의 옵션이 없으면 나중에 자동차나 로봇이 아래로 떨어지게 된다.



- 두 번째 도로 모델링 명령어를 추가한 후, 아래와 같이 2번 항목 (코너1을) 를 선택한다.



● 도로 모델 추가하기 rd1 ^ X

도로 종류:Lane1 X

충돌체 기능 가지기:true X

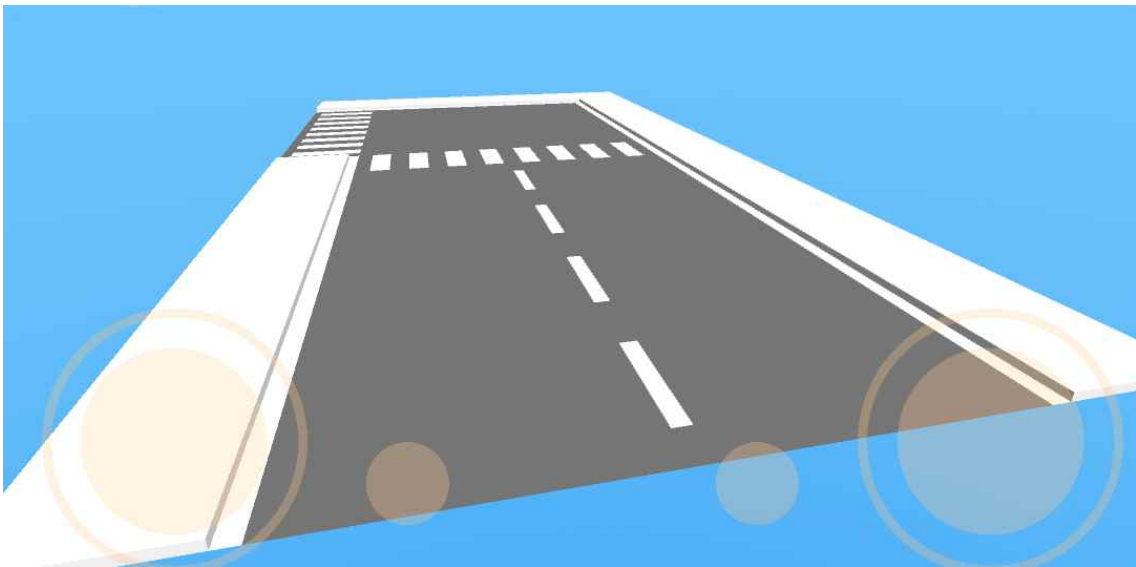
→ 도로 모델 추가하기 rd2 ^ X

도로 종류:Corner1 X

충돌체 기능 가지기:true X

위치:0, 0, 20 X

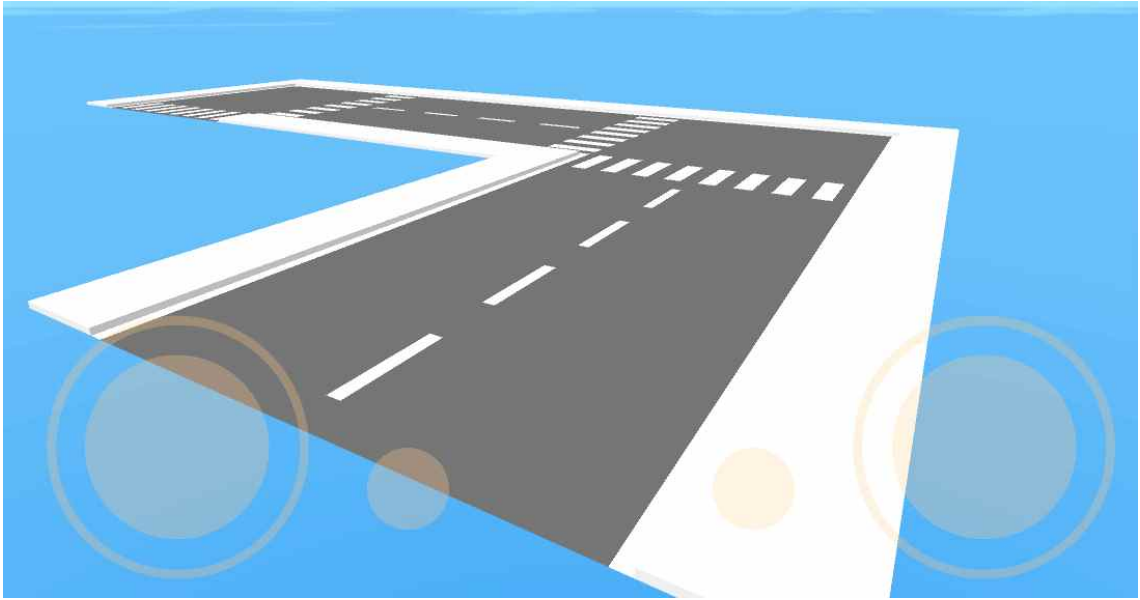
방향:0, -90, 0 X



## 두 번째 코너

- 4각형 도로 트랙이 만들어 지도록 직선 도로와 코너를 추가한다.

방향:0, -90, 0	×
<b>도로 모델 추가하기 rd3</b>	^ ×
도로 종류:Lane1	×
충돌체 기능 가지기:true	×
위치:-20, 0, 20	×
방향:0, 90, 0	×
<b>도로 모델 추가하기 rd4</b>	^ ×
도로 종류:Corner1	×
충돌체 기능 가지기:true	×
위치:-40, 0, 20	×
방향:0, 180, 0	×

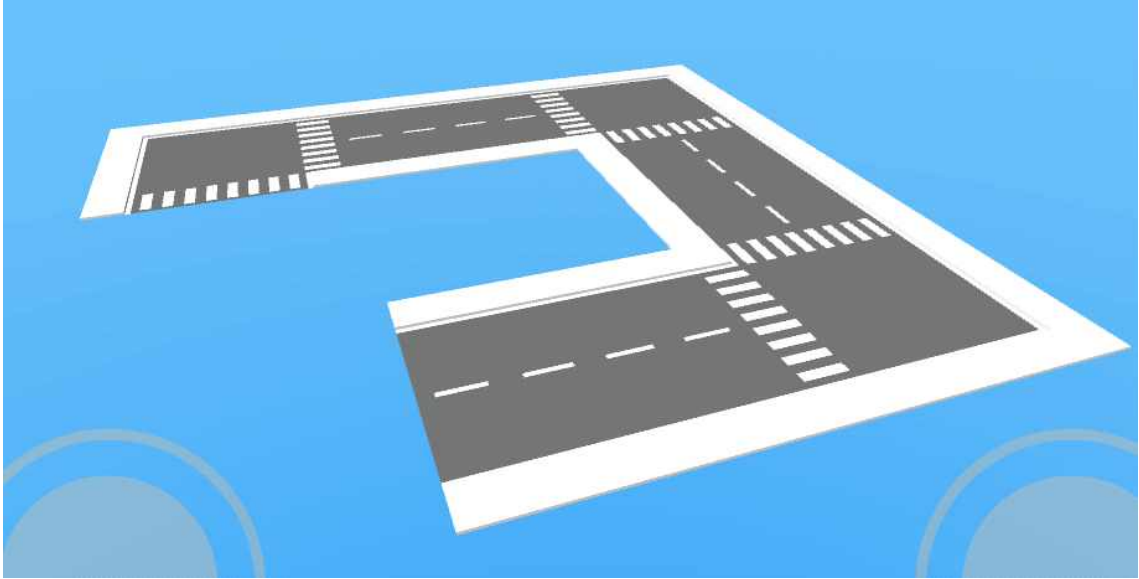




## 세 번째 코너

- 4각형 도로 트랙이 만들어 지도록 직선 도로와 코너를 추가한다.

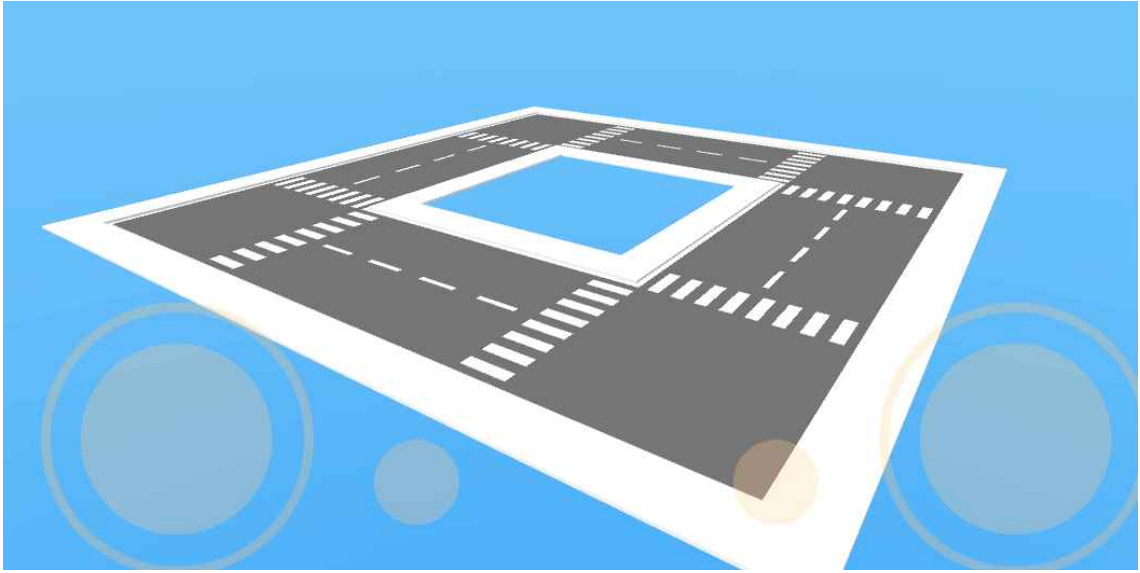




## 네 번째 코너

- 4각형 도로 트랙이 만들어 지도록 직선 도로와 코너를 추가한다.





## 9.2 도로 스크립트 분리하기

### 스크립트 저장

- 현재 만들어진 도로 모델링은 다른 미션에서도 사용되기 때문에 재사용을 위해서 스크립트로 저장한 다음, 다른 프로그램에서 불러오도록 할 수 있다.
- 화면 오른쪽 아래에 있는 저장 버튼을 눌러 파일을 저장한다.



파일명을 입력해 주세요

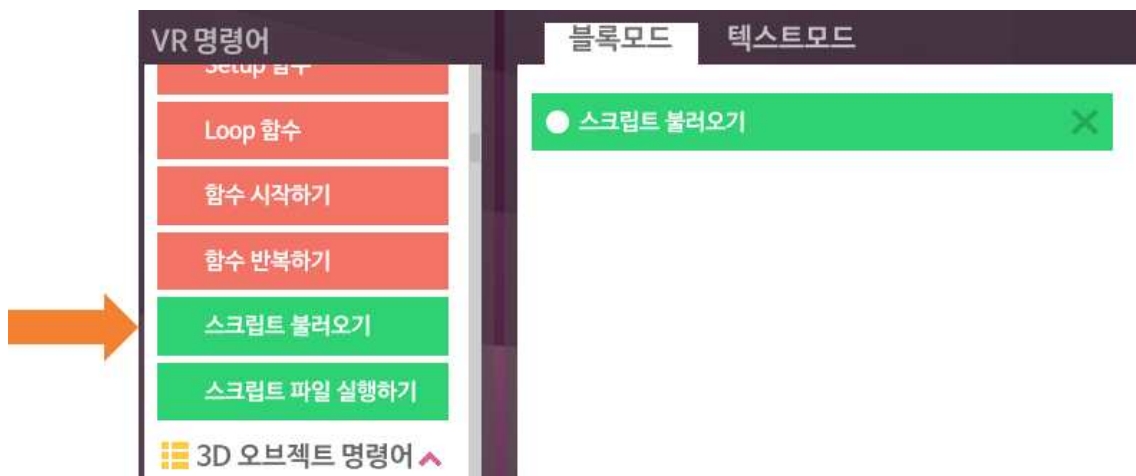
파일 이름

취소 저장

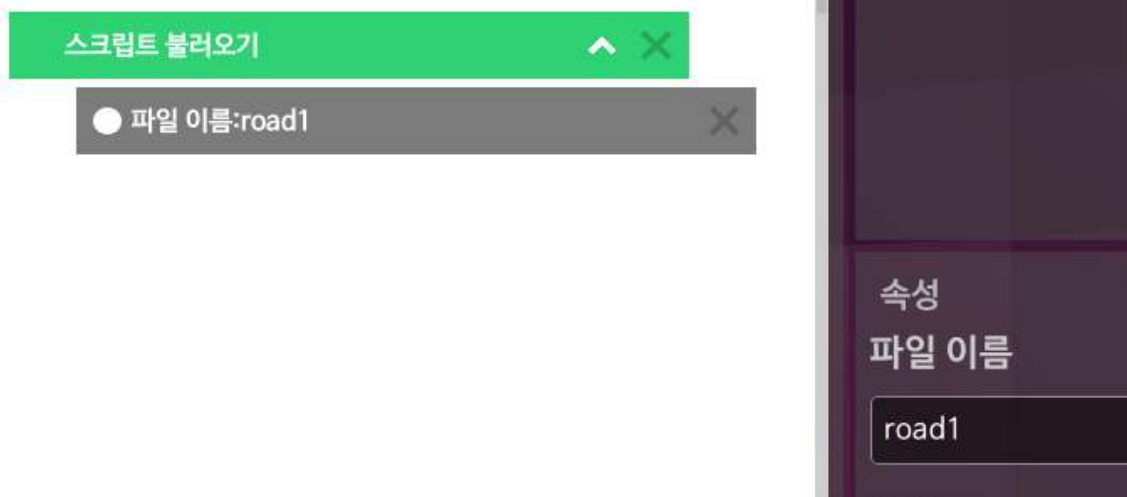
- 파일 이름을 road1으로 저장한다.
- 새파일을 작성한다.

스크립트 불러오기

- 새파일에서 다음과 같이 로직 명령어에 있는 스크립트 불러오기 명령어를 추가한다.



- 옵션에서 파일이름 옵션을 추가한 후, 다음과 같이 방금전에 저장한 road1을 입력해 준다.



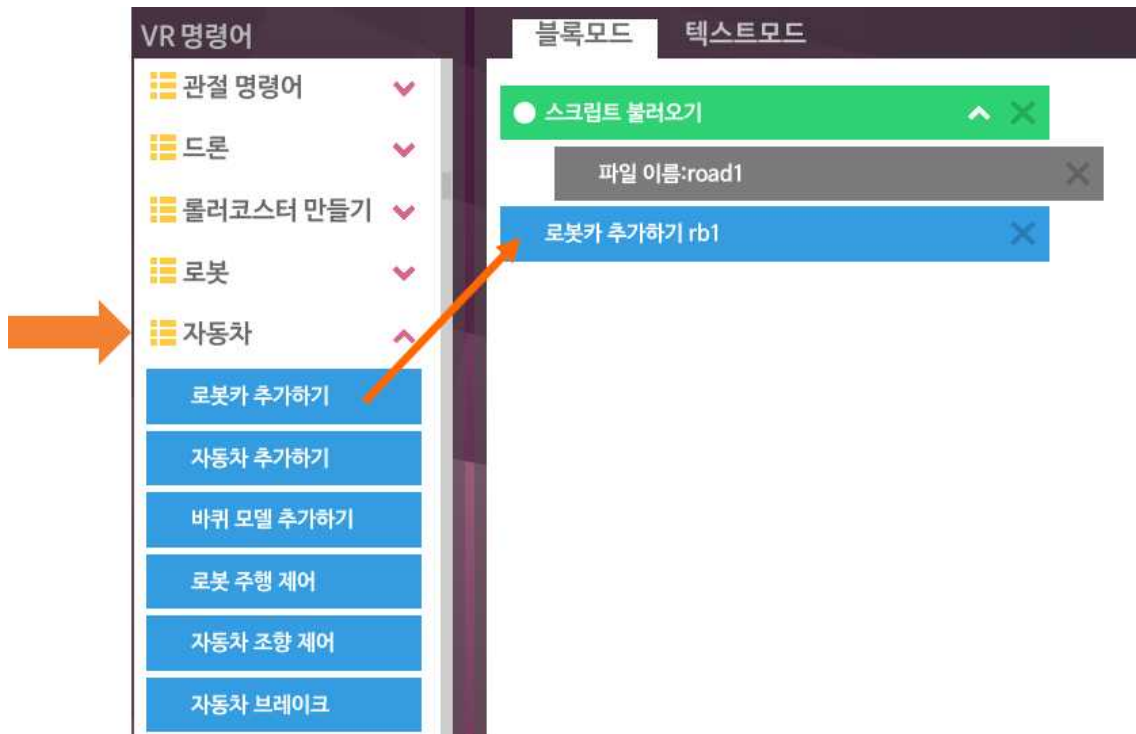
- 프로그램을 실행하면 이전에 작성했던 도로 환경이 그대로 실행되는 것을 알 수 있다.



### 9.3 로봇 카 추가하기

#### 로봇카 추가하기

- 현재 만들어진 도로 모델링은 다른 미션에서도 사용되기 때문에 재사용을 위해서 스크립트로 저장한 다음, 다른 프로그램에서 불러오도록 할 수 있다.





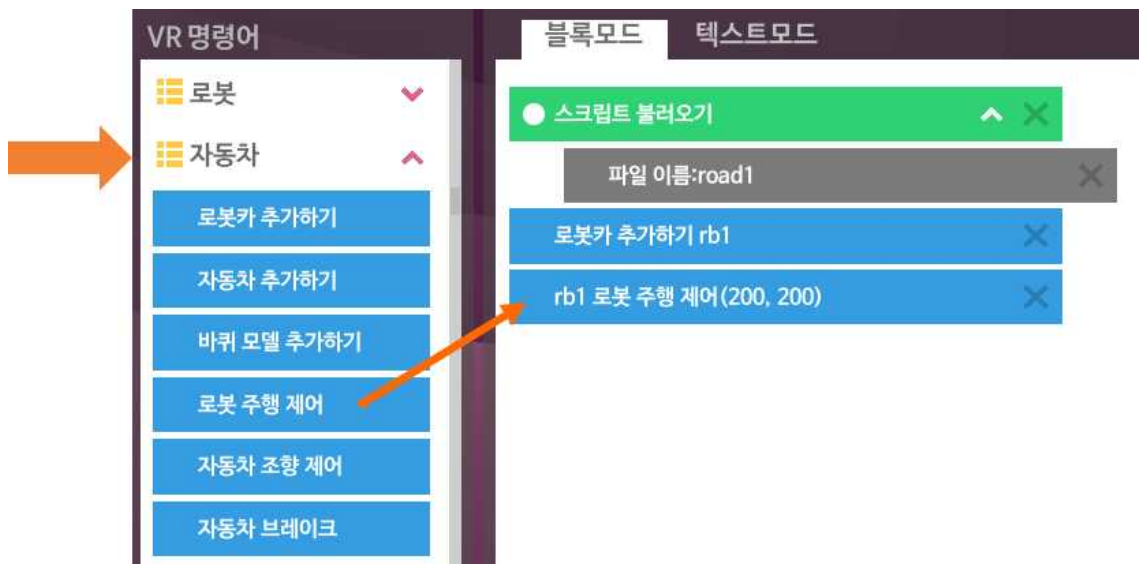
- 실행한 후, 키보드의 방향키를 이용해 로봇카를 조종해 본다.



## 9.4 로봇 카 제어하기

### 로봇 주행 제어

- 로봇카를 제어하기 위해 자동차 명령어 그룹에서 로봇주행 제어 명령어를 아래와 같이 추가한다.



### 로봇 주행 제어

왼쪽	-	200	+	x
오른쪽	-	200	+	x

적용 대상

rb1



## 모터 파워

- 왼쪽과 오른쪽 모터의 파워값의 범위는  $-255 \sim 255$  이다. 즉, 255 값을 지정하면 가장 빠른 파워로 전진하며, -값을 지정할 경우, 반대로 회전하게 된다.

## 파워 비율 증가시키기

- 로봇카의 옵션 중에서 모터 파워 비율을 증가시키면 더 빠른 속도로 로봇카가 이동한다.



후진하기

---

- 아래의 명령어는 후진하는 명령어이다.

● rb1 로봇 주행 제어(-200, -200) X

회전하기

---

- 제자리에서 회전하는 명령어는 다음과 같다.

● rb1 로봇 주행 제어(200, -200) X

1초간 전진, 후진, 회전 반복

- 1초간 전진, 후진, 회전을 반복하려면 다음과 같이 loop 함수와 기다리기 명령어를 이용한다.

The image shows a script editor interface with several stacked blocks. At the top is a green block labeled '스크립트 불러오기' (Load script). Below it is a grey block '파일 이름:road1' (File name: road1). Then a blue block '로봇카 추가하기 rb1' (Add robot car rb1). Below that is another grey block '모터 파워 비율:2' (Motor power ratio: 2). The main part of the script is a red-bordered block labeled '함수 loop()' (Function loop()). Inside this block, there are six lines of code: a blue block 'rb1 로봇 주행 제어(250, 250)' (rb1 robot drive control (250, 250)), a teal block '기다리기(1000)' (Wait (1000)), a blue block 'rb1 로봇 주행 제어(-200, -200)' (rb1 robot drive control (-200, -200)), a teal block '기다리기(1000)' (Wait (1000)), a blue block 'rb1 로봇 주행 제어(150, -150)' (rb1 robot drive control (150, -150)), and a teal block '기다리기(1000)' (Wait (1000)). Each block has a small 'X' icon in the top right corner for deletion.



## 9.5 로봇 카 조종기 구현하기

조이스틱 추가하기

- 로봇카 조종을 위해 loop 함수를 삭제하고 조이스틱과 함수를 추가한다.



The image shows a Scratch script editor with the following blocks:

- 스크립트 불러오기** (Green block)
- 파일 이름:road1** (Grey block)
- 로봇카 추가하기 rb1** (Blue block)
- 모터 파워 비율:2** (Grey block)
- 조이스틱 추가하기** (Green block)
- 오른쪽 조이스틱 함수:f1** (Grey block)
- 함수 void f1()** (Red block)



## 제어 수식

- 조이스틱 명령어로 부터 로봇카 제어 명령어를 계산해 내기 위해 다음과 같은 수식을 함수 안에 추가한다.

The image shows a screenshot of a code editor with several lines of code. The code is organized into a function definition. The first line is a comment in Korean: "조이스틱 추가하기" (Add joystick). The second line is another comment: "오른쪽 조이스틱 함수:f1" (Right joystick function:f1). The third line is the function signature: "함수 void f1 ()". The fourth and fifth lines are assignments: "left = value.X - value.Y" and "right = -value.X - value.Y". The "right" line has a radio button selected. Each line has a close button (X) on the right side.

```
조이스틱 추가하기  
오른쪽 조이스틱 함수:f1  
함수 void f1 ()  
left = value.X - value.Y  
● right = -value.X - value.Y
```

## 로봇카 제어 명령어 추가

- 로봇카 제어 명령어를 함수 안에 추가한다.

스크립트 불러오기 ^ ×

파일 이름:road1 ×

로봇카 추가하기 rb1 ^ ×

모터 파워 비율:2 ×

조이스틱 추가하기 ^ ×

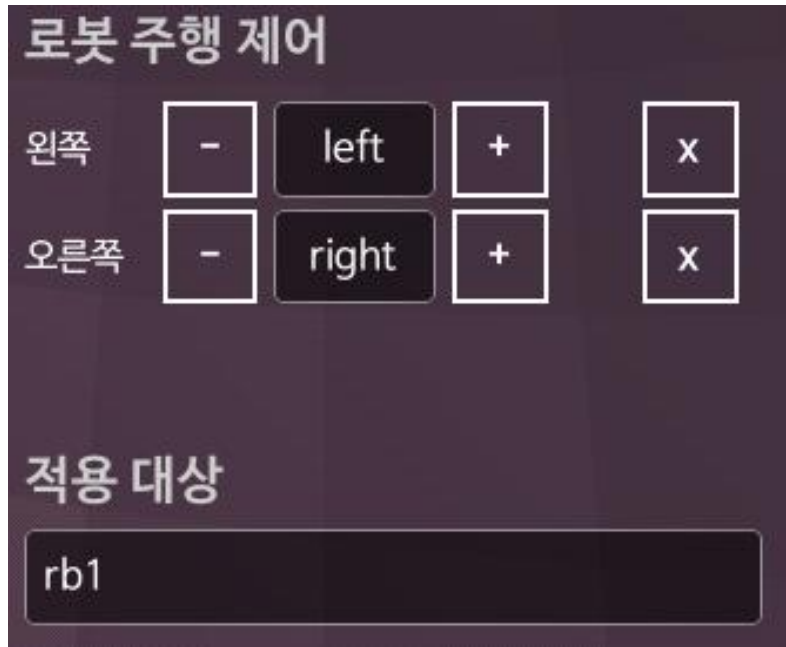
오른쪽 조이스틱 함수:f1 ×

함수 void f1 () ×

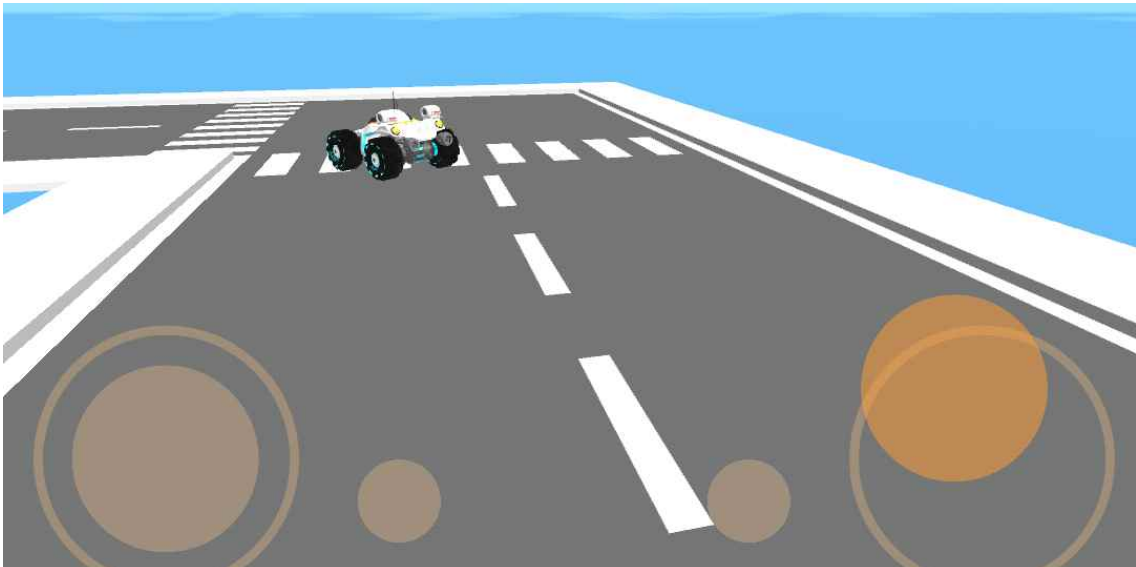
left = value.X - value.Y ×

right = -value.X - value.Y ×

● rb1 로봇 주행 제어(left, right) ×



- 오른쪽 조이스틱을 움직여서 로봇카를 조종해 보기 바란다.



## 9.6 자동차 추가하기

자동차 추가하기

- 새파일을 작성한 후, 도로 환경을 불러온다.



- 자동차 추가하기 명령어를 추가한다.



- 옵션에서 자동차 종류를 추가한다.



- 키보드로 차량을 조종해 본다.



## 9.7 자동차 제어하기

자동차 제어하기

- 자동차를 제어하는 명령어는 자동차 조향 제어 명령어이다. 첫 번째 인수는 앞 바퀴 각도이고, 두 번째 인수는 모터 속도이다.



회전하기

- 다음과 같이 명령어를 수정하여 자동차가 어떻게 움직이는 지 확인해 본다.



## 9.8 조이스틱 조종기 만들기

조이스틱 추가하기

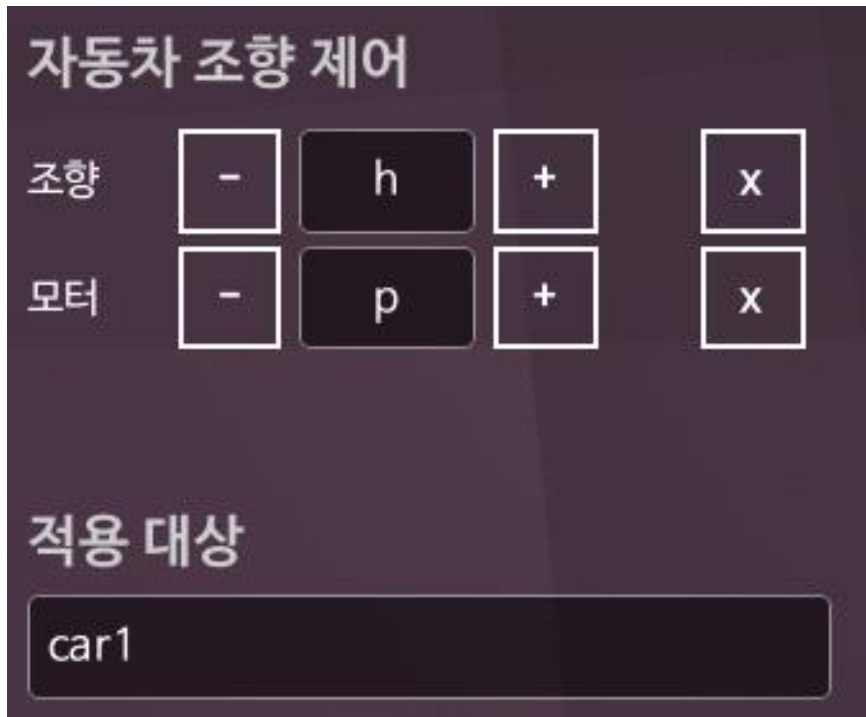
- 자동차를 제어하는 기능을 구현하기 위해 조이스틱과 함수를 각각 추가한다.





- 함수 내부에 자동차를 제어하기 위한 명령어를 다음과 같이 추가한다.

The image shows a script editor interface with several colored blocks representing different parts of a script. At the top is a green block labeled '스크립트 불러오기' (Load script) with an upward arrow and a close button. Below it is a grey block '파일 이름:road1' (File name: road1) with a close button. Next is a blue block '자동차 추가하기 car1' (Add car car1) with an upward arrow and a close button. Below that is a grey block '자동차 종류:Truck1' (Car type: Truck1) with a close button. Then a green block '조이스틱 추가하기' (Add joystick) with an upward arrow and a close button. Below it is a grey block '오른쪽 조이스틱 함수:f1' (Right joystick function: f1) with a close button. The main part of the script is a red block '함수 void f1 ()' (Function void f1 ()) with a close button. Inside this function block are three lines of code: a green block 'h = value.X \* 0.5', another green block 'p = value.Y \* -1', and a blue block 'car1 자동차 조향 제어(h, p)'. Each line has a close button on its right side.



- 오른쪽 조이스틱을 움직여서 자동차를 제어해 본다.



## 실습

---

- 다른 종류의 자동차로 조종해 본다.
- 도로에 다양한 장애물을 추가해 본다.