

[아두이노 시뮬레이션 코딩]

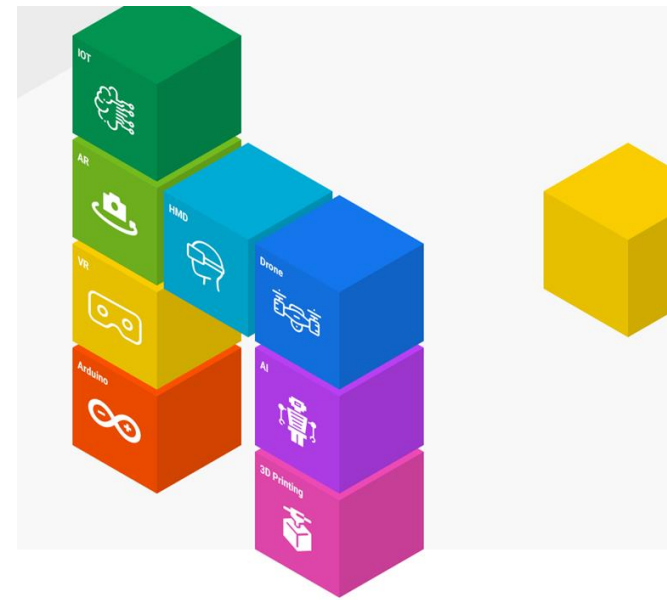
장애물 탐색로봇 만들기



[www.helloapps.co.kr](http://www.helloapps.co.kr)

김 영 준 / 070-4417-1559 / [splduino@gmail.com](mailto:splduino@gmail.com)

# 임의의 장애물 생성하기



# 임의의 장애물 생성하기



random 함수로  
임의의 위치를 지정합니다.

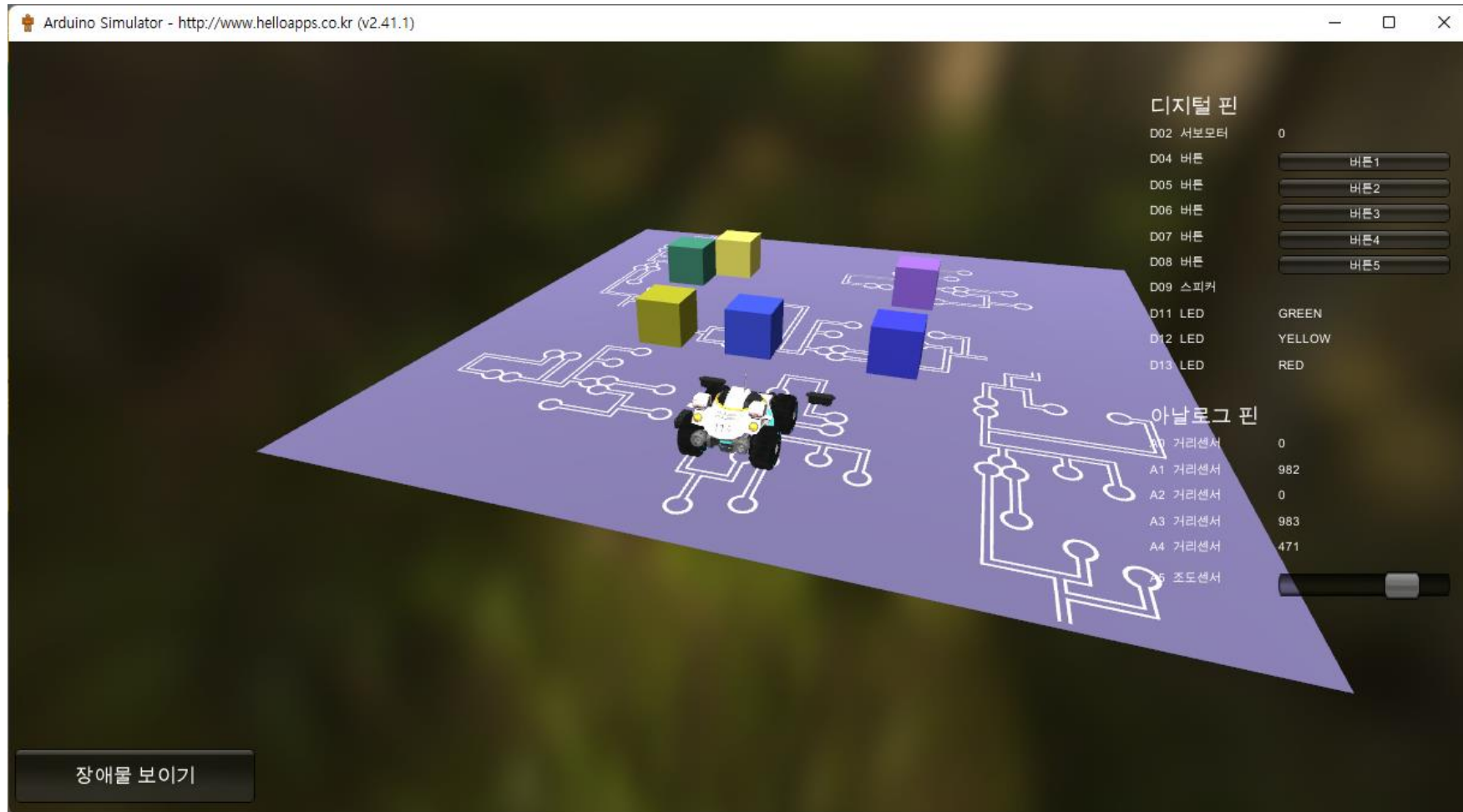
Function Setup

- ☒ Expression HideObstacle()
- for i 를 0 부터 5 까지 i++ 실행
  - ☒ Expression `x = random(-5, 6)`
  - ☒ Expression `z = random(-5, 6)`
  - ☒ Expression `AddCube(x, 0.5, z)`

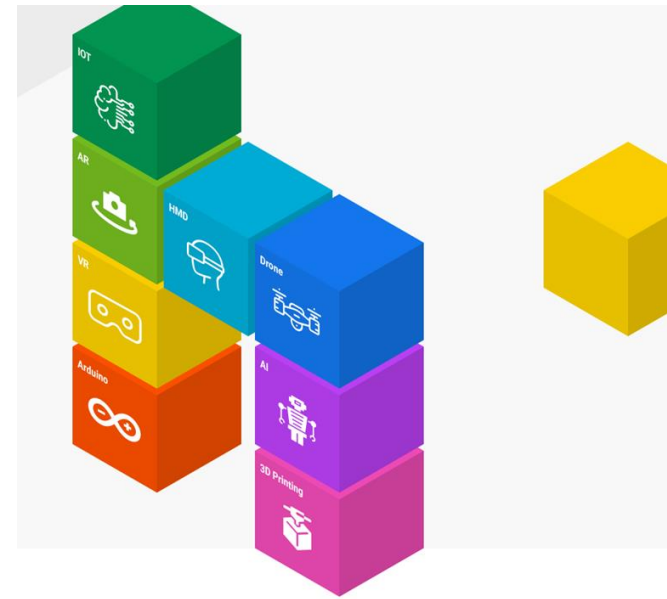
Function Loop

왼쪽에 있는 명령어 아이콘을  
이곳에 마우스로 드래그하여 넣어 주세요.

# 임의의 장애물 생성하기



# 로봇의 위치를 가운데로 옮기기



# 로봇의 위치를 가운데로 옮기기



로봇의 위치를 가운데로  
지정합니다.

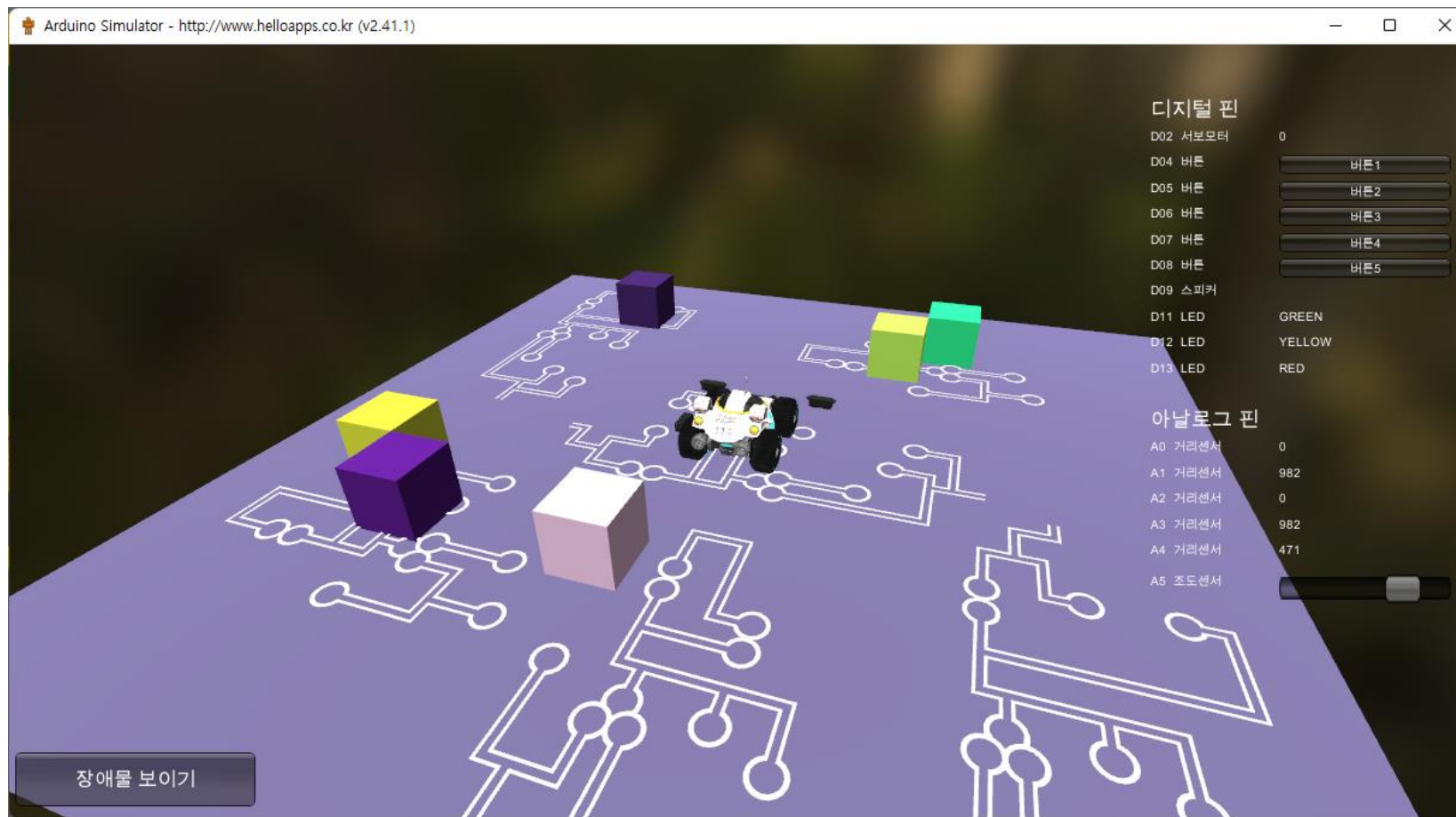
Function Setup

- ☒ Expression HideObstacle()
- for i 를 0 부터 5 까지 i++ 실행
  - ☒ Expression  $x = \text{random}(-5, 6)$
  - ☒ Expression  $z = \text{random}(-5, 6)$
  - ☒ Expression AddCube(x, 0.5, z)
- ☒ Expression SetRobot(0, 0.3, 0)

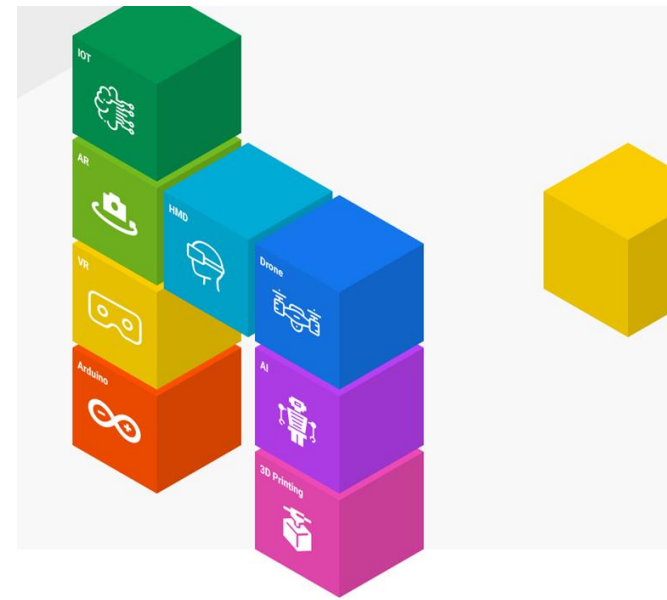
Function Loop

왼쪽에 있는 명령어 아이콘을  
이곳에 마우스로 드래그하여 넣어 주세요.

# 로봇의 위치를 가운데로 옮기기



# 전방 장애물 감지하기

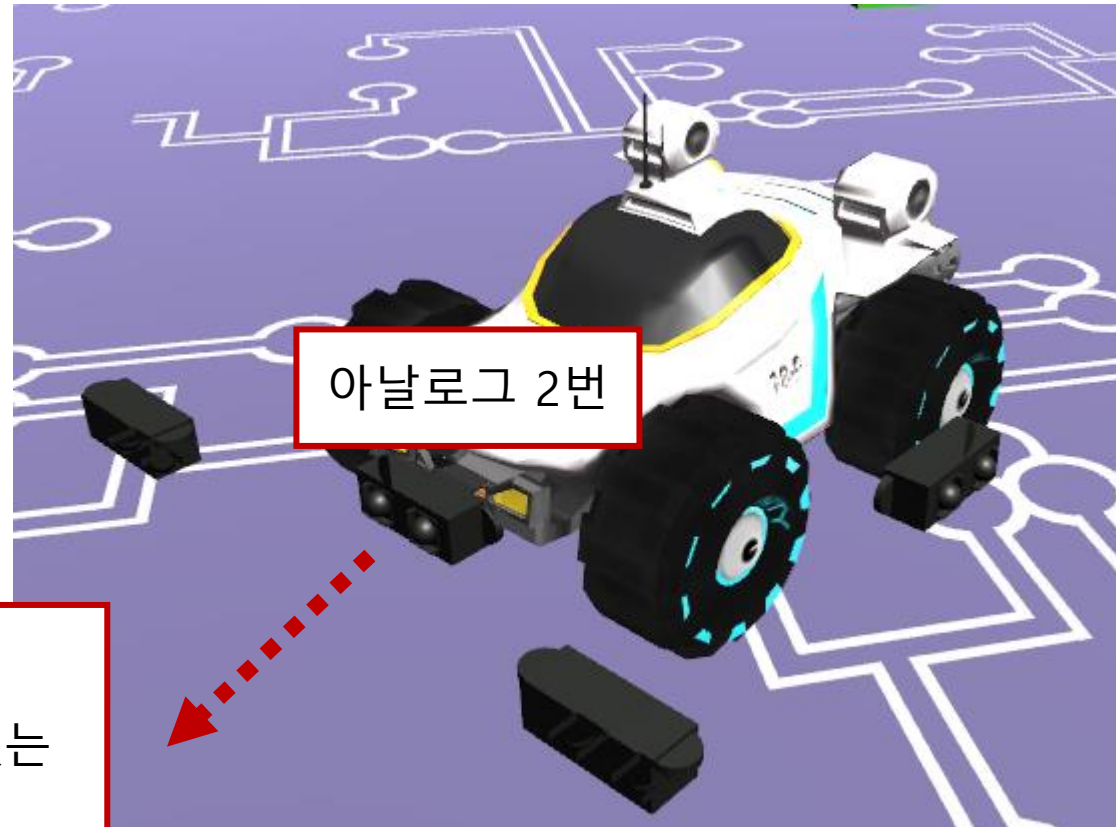




# 자율주행 로봇카의 구성



아날로그 2번 PSD 거리 센서는 로봇카의 앞쪽에 있는 물체까지의 거리를 측정



아날로그 2번

아날로그 2번 센서는  
로봇카의 앞쪽 방향에 있는  
물체까지의 거리 측정

# 로봇을 회전시켜 장애물을 탐색

로봇을 제자리 회전시킴

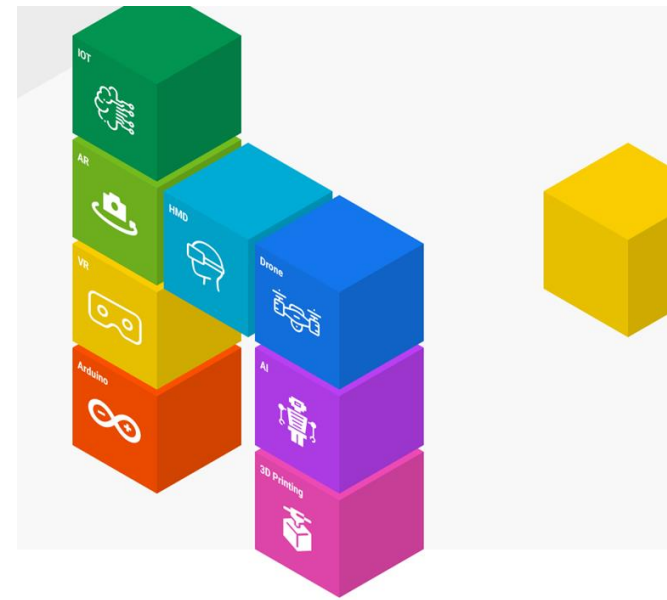
전방 센서의 값을  
출력시킴

```
Function Setup
  [x] Expression HideObstacle()
  for i 를 0 부터 5 까지 i++ 실행
    [x] Expression x = random(-5, 6)
    [x] Expression z = random(-5, 6)
    [x] Expression AddCube(x, 0.5, z)
  [x] Expression SetRobot(0, 0.3, 0)
  [x] DriveWrite 50 (Left) -50 (Right)

Function Loop
  [x] a2 = AnalogRead 2 (핀번호)
  [x] PrintLine a2
  [x] Delay 10 (밀리초)
```



# 전방 장애물 떨어트리기



# 전방 장애물 떨어트리기

회전하다가 전방에  
장애물을 만나면  
앞으로 전진합니다.

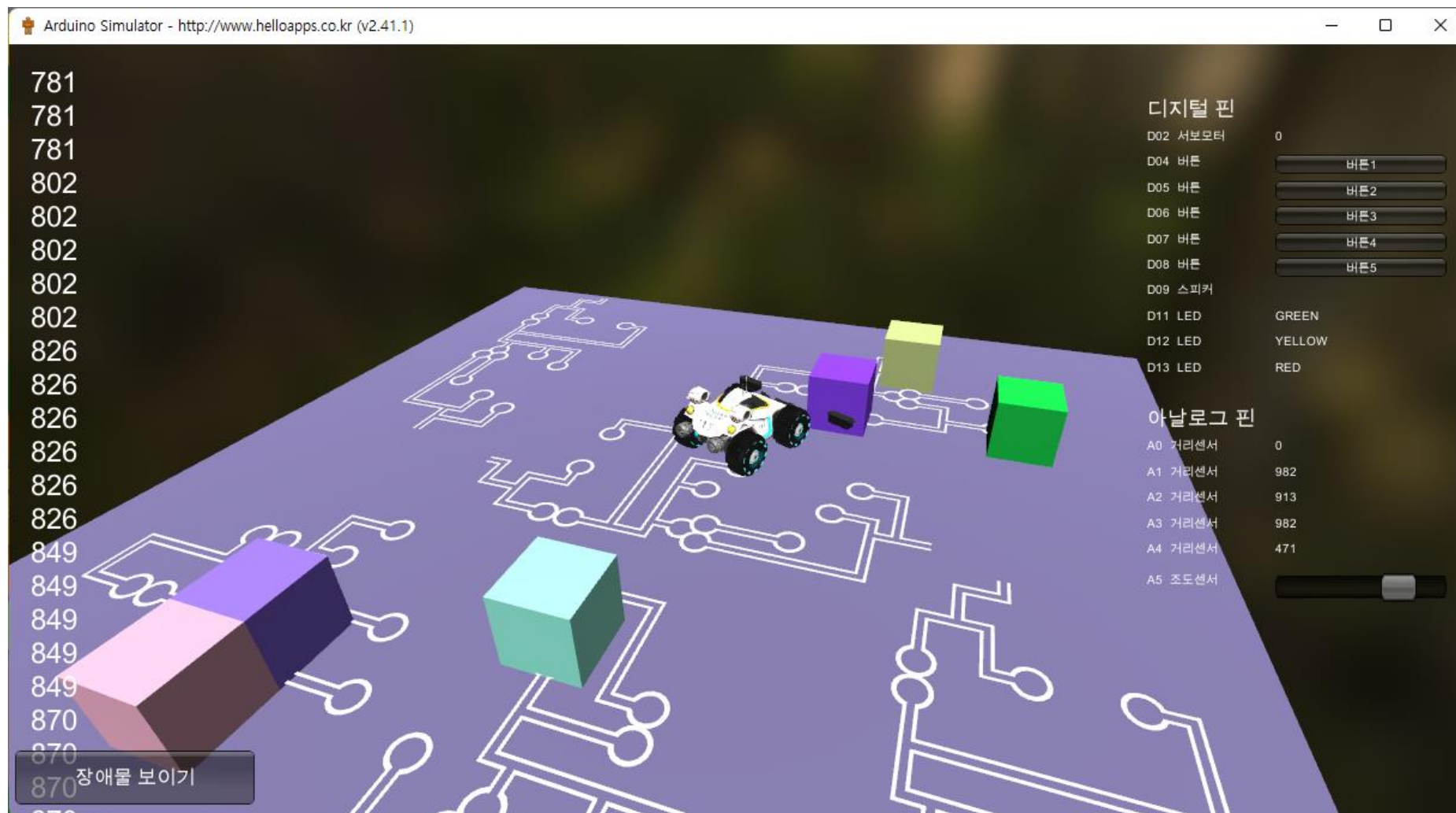
Function Setup

- Expression: HideObstacle()
- for i 를 0 부터 5 까지 i++ 실행
  - Expression:  $x = \text{random}(-5, 6)$
  - Expression:  $z = \text{random}(-5, 6)$
  - Expression: AddCube(x, 0.5, z)
- Expression: SetRobot(0, 0.3, 0)
- DriveWrite: 50 (Left) -50 (Right)

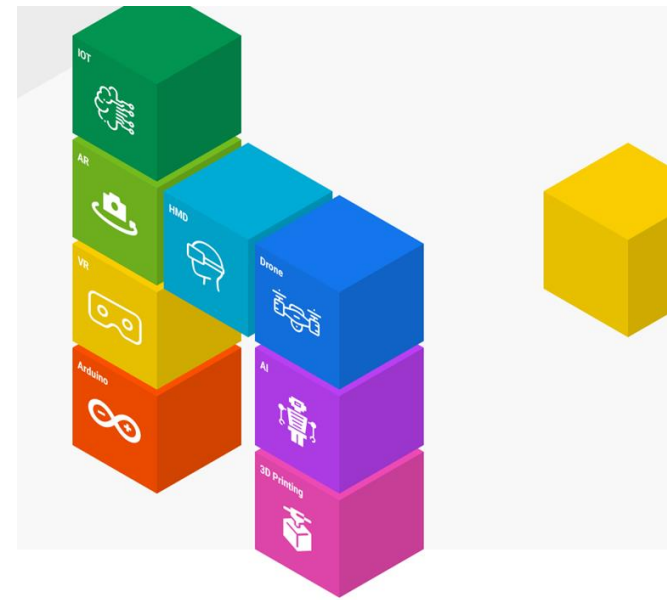
Function Loop

- a2 = AnalogRead 2 (핀번호)
- PrintLine: a2
- if  $a2 > 0$ 
  - DriveWrite: 100 (Left) 100 (Right)
- Delay: 10 (밀리초)

# 전방 장애물 떨어트리기



# 떨어지지 않는 기능 추가하기



# 떨어지지 않는 기능 추가하기



1번 센서의 값을  
출력해 봅니다.

Function Loop

- ☒ a1 = AnalogRead 1 (핀번호)
- ☒ a2 = AnalogRead 2 (핀번호)
- ☒ PrintLine a1
- if a2 > 0
  - ☒ DriveWrite 100 (Left) 100 (Right)
- ☒ Delay 10 (밀리초)





# 떨어지지 않는 기능 추가하기



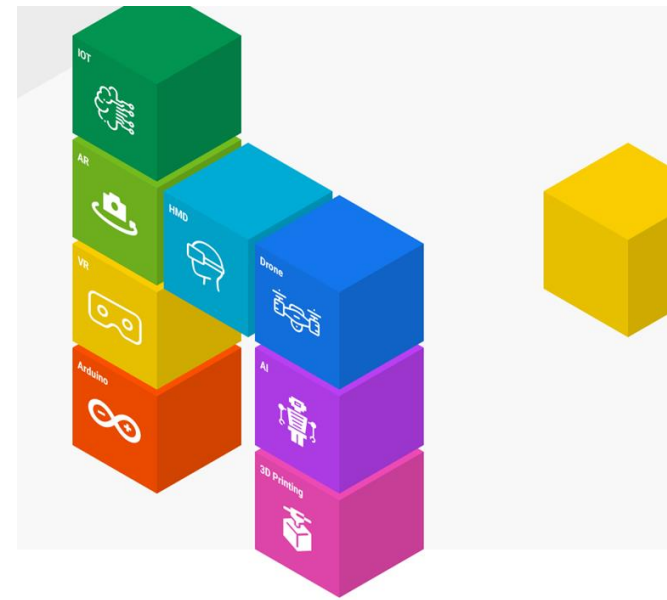
떨어지지 않게 하는  
기능을 추가합니다.

Function Loop

- ☒ a1 = AnalogRead 1 (핀번호)
- ☒ a2 = AnalogRead 2 (핀번호)
- ☒ PrintLine a1
- if a1 < 100
  - ☒ DriveWrite 50 (Left) -50 (Right)
- if a2 > 0
  - ☒ DriveWrite 100 (Left) 100 (Right)
- ☒ Delay 10 (밀리초)

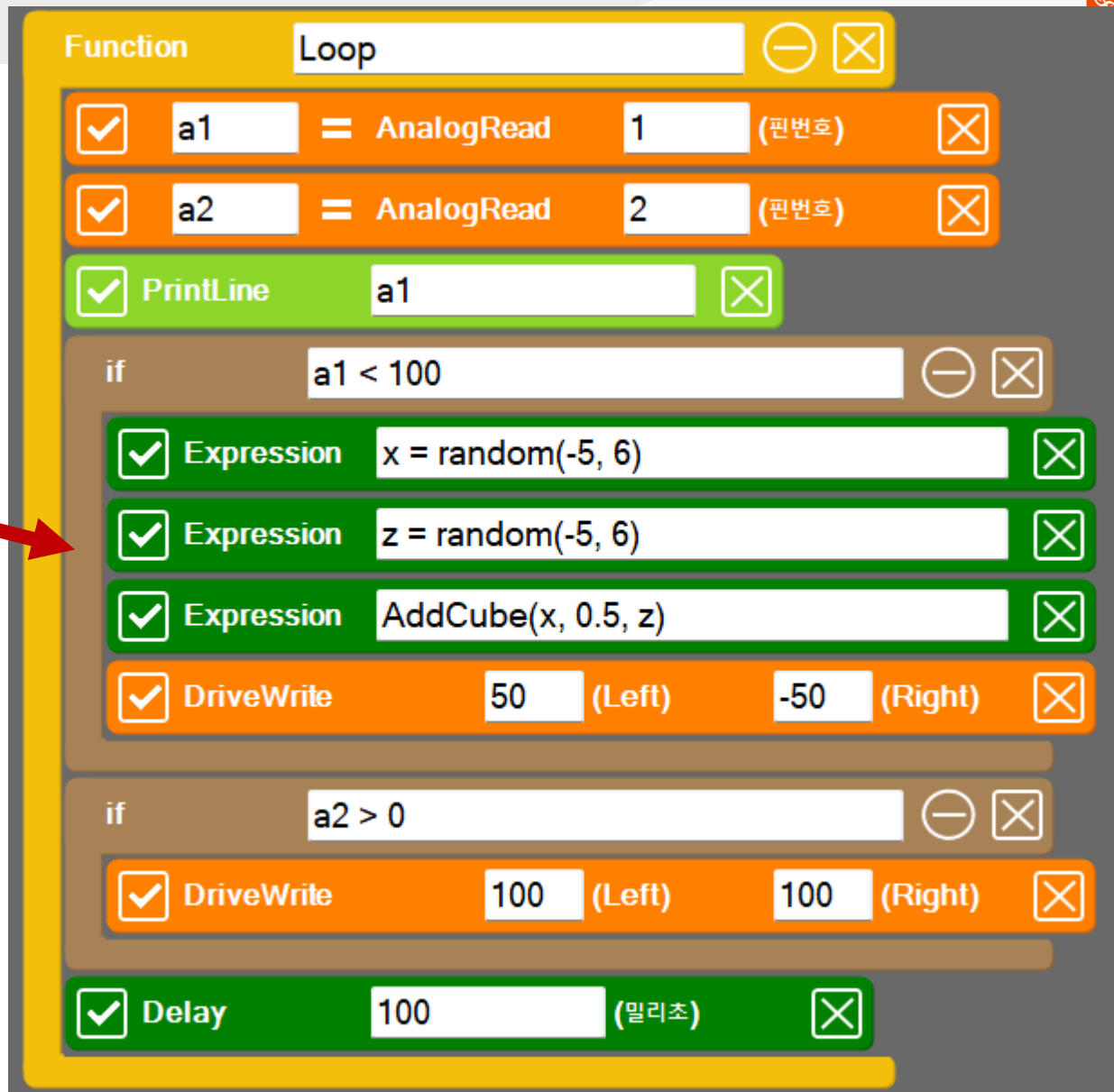


# 장애물 자동 추가 기능



# 장애물 자동 추가 기능

임의의 위치에  
장애물을 생성하는  
기능을 추가합니다.



The image shows a Scratch script titled "Function Loop". The script contains the following blocks:

- Function Loop** (yellow block with minus and close buttons)
- Checkmark** **a1** **=** **AnalogRead** **1** **(핀번호)** **X** (orange block)
- Checkmark** **a2** **=** **AnalogRead** **2** **(핀번호)** **X** (orange block)
- Checkmark** **PrintLine** **a1** **X** (green block)
- if** **a1 < 100** **X** (brown block with minus and close buttons)
  - Checkmark** **Expression** **x = random(-5, 6)** **X** (green block)
  - Checkmark** **Expression** **z = random(-5, 6)** **X** (green block)
  - Checkmark** **Expression** **AddCube(x, 0.5, z)** **X** (green block)
  - Checkmark** **DriveWrite** **50** **(Left)** **-50** **(Right)** **X** (orange block)
- if** **a2 > 0** **X** (brown block with minus and close buttons)
  - Checkmark** **DriveWrite** **100** **(Left)** **100** **(Right)** **X** (orange block)
- Checkmark** **Delay** **100** **(밀리초)** **X** (green block)

