



대회 미션 실습

지금까지 배운 내용으로 인공지능 로봇 자율 미션을 진행해봅니다.

이곳에서 미션을 연습해봅시다.

제일 먼저 라이브러리를 가져옵니다.

In []:

```
from zumi.zumi import Zumi
from zumi.util.screen import Screen
from zumi.personality import Sound
from zumi.util.camera import Camera
from zumi.util.vision import Vision
from zumi.protocol import Note
from zumi.util.color_classifier import ColorClassifier
import time
```

```
zumi = Zumi()
screen = Screen()
camera = Camera()
vision = Vision()
sound = Sound(zumi)
```

챌린지 설명

챌린지 A

- ① 경기 전 룰렛 돌려 장소를 선택 받습니다.
- ② 선택받은 장소의 QR코드를 준비하여 경기 시작 후 로봇에 QR코드를 가져가 인식 시켜줍니다
- ③ 로봇은 선택받은 장소의 QR코드를 인식하였다는 것을 알려주는 도,레,미 소리를 내야 합니다.

-

카메라와 **find_QR_code()** 함수를 사용하여 QR 코드 메시지 읽습니다.
메시지를 읽었다면 도레미 음계 소리를 냅니다.

In []:

```
#####
##### Challenge A - 1 #####
#####
#                                                                 #
# QR 카드를 인식하여 이동할 위치를 알아냅니다.                  #
#                                                                 #
#####

try:
    #### QR 코드를 읽습니다.

    # 이곳에 코드를 작성하세요.


    #### QR 코드를 찾았다면 소리내기

    # 이곳에 코드를 작성하세요.


# ■ 인터럽트(정지) 버튼을 누르면 정지되도록 함.
except KeyboardInterrupt:
    zumi.stop()
    camera.close()
    print("인터럽트 버튼을 눌렀으므로 정지합니다.")

finally:
    zumi.stop()
    camera.close()
    print("코드 종료")
```

A-2

블록을 세면서 해당 건물이 위치한 블록 앞까지 이동

QR 메시지로 알아낸 위치로 **drive_over_markers()** 함수를 사용하여 이동합니다.

In []:

```
#####
##### Challenge A - 2 #####
#####
#                                                                 #
# QR 메시지로 알아낸 위치로 이동하여 주차합니다.                #
#                                                                 #
#####

##                ##                |#####|
##                ##                |#|#####|##|#####|
##                ##                |#|                ##|#####|
##                ##                |#|                ##|#####|
##                ##                |#|    building    ##|                ##|    factory
##                ##                |#|                ##|#####|
##                ##                |#|                ##|#####|
##                ##                ##                ##|#####|
##                #####                #####
##
##                [] [] [] []        [] [] [] []        [] [] [] []        [] [] []
##                [] [] [] []        [] [] [] []        [] [] [] []        [] [] []
##                [] [] [] []        [] [] [] []        [] [] [] []        [] [] []
##                [] [] [] []        [] [] [] []        [] [] [] []        [] [] []
##                [] [] [] []        [] [] [] []        [] [] [] []        [] [] []
##
#####                #####|#####|
##|                |##                ##|                ##|
##|                |##                ##|                ##|
##|                |##                ##|                ##|
##|    museum    |##                ##|    school    ##|
##|                |##                ##|                ##|
##|                |##                ##|                ##|
##|#####|##|                ##|#####|
#####                #####
```

```
try:
    #### 블록을 세면서 해당 건물이 위치한 블록 앞까지 이동합니다.

    # 이곳에 코드를 작성하세요.
```

```
# ■ 인터럽트(정지) 버튼을 누르면 정지되도록 함.
except KeyboardInterrupt:
    zumi.stop()
    camera.close()
    print("인터럽트 버튼을 눌렀으므로 정지합니다.")

finally:
    zumi.stop()
    camera.close()
    print("코드 종료")
```

해당 건물의 주차위치로 진입한 후 빠져나오기

QR 메시지의 건물에 따라 목적지로 진입합니다.
지정된 장소에 주차한 후에 다시 도로로 빠져나옵니다.

In []:

[illegible]

```
try:
    ##### 해당 건물의 주차위치로 진입합니다.
```

이곳에 코드를 작성하세요.

도로로 복귀하기

이곳에 코드를 작성하세요.

```
# ■ 인터럽트(정지) 버튼을 누르면 정지되도록 함.
except KeyboardInterrupt:
    zumi.stop()
    camera.close()
    print("인터럽트 버튼을 눌렀으므로 정지합니다.")
```

finally:

```
zumi.stop()  
camera.close()  
print("코드 종료")
```

A-4

정지 표시 인식 코스로 이동하기

도로를 따라서 정지 표시 인식 코스로 이동합니다.

갈때기 구간에 도달하면 라인 팔로워(**line_follow_gyro_assist**) 함수 혹은 갈때기 통과(**funnel_align**) 함수를 사용하여 정지표지판 앞에 정지합니다.

In []:

```
#####
##### Challenge A - 4 #####
#####
#
# 정지 표시 인식 코스로 이동하기
#
#####
```

[illegible]

```
try:
    ##### 다음 코스로 이동하기

    # 이곳에 코드를 작성하세요.


    ##### 챌린지 A 종료

# ■ 인터럽트(정지) 버튼을 누르면 정지되도록 함.
except KeyboardInterrupt:
    zumi.stop()
    camera.close()
    print("인터럽트 버튼을 눌렀으므로 정지합니다.")

finally:
    zumi.stop()
    camera.close()
    print("코드 종료")
```


챌린지 B

Challenge A 이후로 연결 됩니다. 로봇은 북쪽을 바라보고 있습니다.

해당 챌린지의 목표는 로봇에게 보여진 이미지를 인식하고 미로의 벽을 인식하여 미로 벽에 충돌하지 않으며 다음 장소까지 주행 하는 것이 이 챌린지의 목표입니다.

- ① 로봇이 STOP 카드 앞에서 멈춰 섭니다
(여기에서 STOP 카드는 위 이미지 처럼 표시된 부분에 위치해야 합니다)
- ② 로봇이 STOP 카드를 인식했다는 도,레,미 소리를 냅니다.
- ③ 소리가 끝나면 재조정을 할 것 인지 선택 할 수 있습니다. (로봇을 만지면 재조정으로 판단)
-> 재조정을 했다면 : 재조정이 끝나면 STOP 카드가 제거될 때까지 기다려야합니다.
-> 재조정을 안했다면 : STOP 카드를 제거 합니다.
(로봇이 움직이는 순간부터는 재조정 절대 할 수 없습니다. 위반시 미션종료)
- ④ STOP 카드가 제거되면, 로봇은 도로를 주행 합니다.
- ⑤ 로봇은 벽에 부딪히지 않으며 Challenge C 방향으로 통과구간 1,2,3을 지나도록 주행합니다.
(통과 구간을 지날 때에는 로봇의 몸체가 완전히 다 통과 해야 점수로 인정 됩니다,)



B-1

정지 표지판을 찾고, 정지표지판이 사라지면 이동하기

카메라와 `detect_stop_sign()` 함수를 사용하여 정지 표시를 찾습니다.

정지 표시를 찾으면 도레미 소리를 냅니다.

정지 표시가 사라질때까지 기다립니다.

이 구간에는 주미를 잡아서 위치를 재조정 할 수 있습니다.

재조정이 가능한 시점은 정지표지판을 인식 하고 "도레미"소리가 났을 때입니다.

심판에게 재조정하겠다는 의사를 표시하고 확인을 받은 후에 진행합니다.

재조정을 하는 경우 프로그램을 나누어 작성하고 다음 셀을 실행하게 됩니다.

재조정을 하지 않도록 작성하는 경우에는 하나의 셀에 모든 프로그램이 작성됩니다.

In []:

```
#####
##### Challenge B -1 #####
#####
#
# 정지 표지판을 찾고, 정지표지판이 사라지면 이동하기
#
#####
```

[illegible]

```
try:
```

```
#### 정지 표지판을 찾기

# 이곳에 코드를 작성하세요.
```

```
#### 정지 표지판을 찾으면 소리내기
# 이곳에 코드를 작성하세요.
```

```
#### 정지 표지판이 사라질때까지 기다리기
# 이곳에 코드를 작성하세요.
```

```
# ■ 인터럽트(정지) 버튼을 누르면 정지되도록 함.  
except KeyboardInterrupt:  
    zumi.stop()  
    camera.close()  
    print("인터럽트 버튼을 눌렀으므로 정지합니다.")  
  
finally:  
    zumi.stop()  
    camera.close()  
    print("코드 종료")
```

B-2

벽을 피해서 이동

벽을 감지하고 원하는 방향으로 회전하여 다음 목적지로 진행합니다.

In []:

[illegible]

```
try:
    #### 벽을 피해서 이동하기

    # 이곳에 코드를 작성하세요.

# ■ 인터럽트(정지) 버튼을 누르면 정지되도록 함.
except KeyboardInterrupt:
    zumi.stop()
    camera.close()
    print("인터럽트 버튼을 눌렀으므로 정지합니다.")

finally:
    zumi.stop()
    camera.close()
    print("코드 종료")
```

B-3

깔때기 구간 집입하고 주미 정지

깔때기 구간에 도달하면 라인 팔로워(line_follow_gyro_assist) 함수 혹은 깔때기 통과(funnel_align) 함수를 사용하여 정지표지판 앞에 정지합니다.

In []:

```
#####
##### Challenge B - 3 #####
#####
#                                                                 #
# 깔때기 구간 집입하고 주미 정지                                #
#                                                                 #
#####

try:
    #### 깔때기 구간 집입하여 주미 정지

    # 이곳에 코드를 작성하세요.


    #### 챌린지 B 종료

# ■ 인터럽트(정지) 버튼을 누르면 정지되도록 함.
except KeyboardInterrupt:
    zumi.stop()
    camera.close()
    print("인터럽트 버튼을 눌렀으므로 정지합니다.")

finally:
    zumi.stop()
    camera.close()
    print("코드 종료")
```

챌린지 C

Challenge B 이후로 연결 됩니다. 로봇은 북쪽을 바라보고 있습니다.

해당 챌린지의 목표는 로봇에게 보여진 색상을 인식하고 주행하며 QR 코드를 읽고 최종 지점으로 주행 하는 것이 이 챌린지의 목표입니다.

- ① 로봇은 컬러 카드 (빨간색) 앞에서 멈춰서며, 빨간 카드는 매트의 표시된 곳에 위치해야 합니다.
- ② 이 때 로봇은 반드시 빨간색을 인식 해야하고 로봇이 색상을 인식 했다는 도,레,미 소리를 냅니다.
(이 소리가 완전히 끝날 때 까지 참가자는 기다려야 합니다.)
- ③ 소리가 끝나면 재조정을 할 것 인지 선택 할 수 있습니다. (로봇을 만지면 재조정으로 판단)
- ④ 재조정이 끝나면 참가자는 컬러 카드(빨간색) 을 컬러 카드 (초록색)으로 바꿉니다.

(색상 카드가 초록색 카드로 바뀌기 전 까지 로봇은 절대 움직여서는 안됩니다.

로봇이 움직이는 순간부터는 재조정 절대 할 수 없으며 이를 위반 하면 미션이 종료 됩니다.)

⑤ 초록색 카드를 인식 하였다면 로봇 선을 따라 주행합니다. (선 이탈 시 패널티)

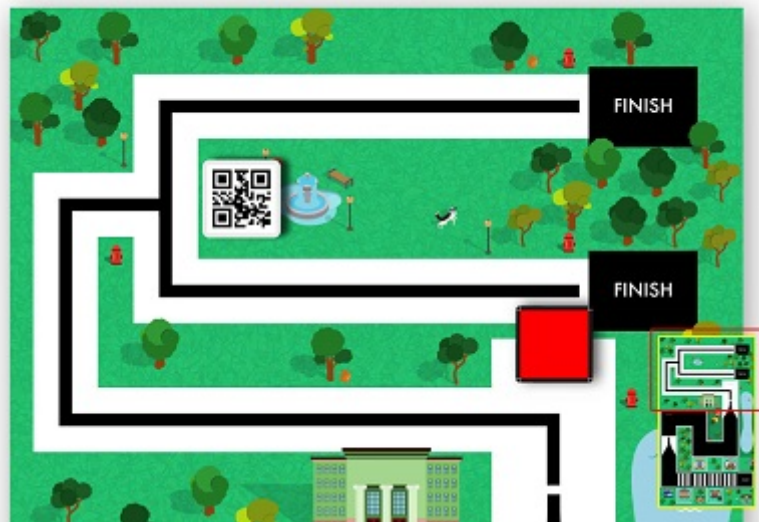
⑥ QR박스가 있는 곳에 도착하면 QR 박스의 QR 코드를 인식 합니다.

(QR박스는 경기 시작 전 룰렛을 돌려 QR방향을 결정 받고 위쪽 'QR 박스 구역' 에 QR을 방향을 맞춘 상태에서 자유롭게 배치하며,반드시 경기 시작전에 미리 배치되어 있어야 합니다.)

⑦ 로봇은 이렇게 배치된 QR박스의 QR 코드를 인식합니다.

⑧ T교차로에서 QR 박스의 해당 방향에 맞게 선을 따라서 FINISH 지점까지 주행 합니다.

⑨ 추가 미션을 수행 하고 추가 미션이 종료되었다면 도,레,미 소리 를 냅니다



챌린지 C 미션을 진행하기 위해서는 색상을 먼저 훈련시켜야 합니다.

데모 - 색상훈련을 통해서 빨간색과 초록색을 미리 훈련시키도록 합니다.

여기서 훈련한 모델을 가지고 미션을 진행합니다.

조명이나 환경에 따라 색상인식률이 달라질 수 있으므로, 실제 맵의 주미가 인식해야할 위치에서 훈련시키는 것이 인식률을 높일 수 있습니다.

C-1

빨간색 카드를 인식하고, 초록색 카드로 바뀌면 이동하기

정지한 상태에서 빨간색 카드를 인식한 후에 도,레,미 소리를 냅니다.

빨간색 카드가 초록색 카드로 변경될때까지 기다립니다.

In []:

```
#####
##### Challenge C - 1 #####
#####
#
# 데모에서 작성한 KNN 모델을 사용하여 색상을 인식합니다.
#
# 색상카드를 읽습니다.
# - 빨강일때 정지
# - 초록색일때 이동
#
#####
```

[illegible][illegible]

```
try:
```

데모에서 미리 작성된 사용자의 KNN 모델을 불러옵니다.
이곳에 코드를 작성하세요.

빨강색을 찾을때까지 기다립니다.
이곳에 코드를 작성하세요.

빨강색을 찾았다면 소리를 냅니다.
이곳에 코드를 작성하세요.

####초록색으로 바꿀 때까지 기다립니다.
이곳에 코드를 작성하세요.

```
finally:  
    zumi.stop()  
    camera.close()
```

C-2

라인을 따라서 QR 박스로 이동하기

라인을 따라서 QR 박스로 이동합니다.

line_follow_gyro_assist() 함수를 사용하여 이동합니다.

- 각각의 주미에 따라 speed, angle_adj, l_th, r_th 값을 알맞은 값으로 조정합니다.

In []:

```
#####
##### Challenge C - 2 #####
#####
#
# 라인을 따라서 QR 박스의 위치까지 이동합니다.
#
#####

#////////////////////////////////////
#
#           #////////////////////////////////
#           # ///////////////
#           # #
#           # #
#           # #
#           # #   # /////#
#   ##/////////////// #   #   #
#   # /////////////// #   #   #
#   # #           # #   #/////#
#   # /           # #
#   # /           # #
#   # /           # #
#   # /           # ///////////////
#   # /           ///////////////
#   # /
#   # ///////////////////////////////////////////////////#
#   ##///////////////////////////////////////////////////
#
#
#
#
# //////////////////////////////////// //   #   #   //
# ###///////////////###///////////////###///////////////## ##   #   #   ##
# //                               /#   /#   #/   /#   ##
# /#                               ##   /#   #/   /#   ##
# /#                               ##   /#   #/   /#   ##
# /#                               ##   /#   #/   /#   ##
```

```
try:
    #### 라인을 따라 이동하기
    # 이곳에 코드를 작성하세요.
```

```
finally:
    zumi.stop()
    camera.close()
```

C-3

QR 박스의 메시지를 읽고 메시지에 따라 이동하기

QR 박스를 읽습니다.

해당 QR의 메시지에 따라서 left , right로 이동합니다.

line_follow_gyro_assist() 함수를 사용하여 이동합니다.

- 각각의 주미에 따라 speed, angle_adj, l_th, r_th 값을 알맞은 값으로 조정합니다.

In []:

[illegible]

```
try:
    ##### QR 코드를 읽고 메시지에 따라서 이동하기
    # 이곳에 코드를 작성하세요.
```

```
#### 종료되었다면 소리내기
# 이곳에 코드를 작성하세요.
```

```
finally:
    zumi.stop()
    camera.close()
```

