

도둑 잡기

D-2 조

김우정 복권근 장세환 최현성

순서

1. OverView
 2. 아키텍처
 3. 모델 및 월드
 4. 알고리즘 및 어려웠던 점
 5. 시연
-

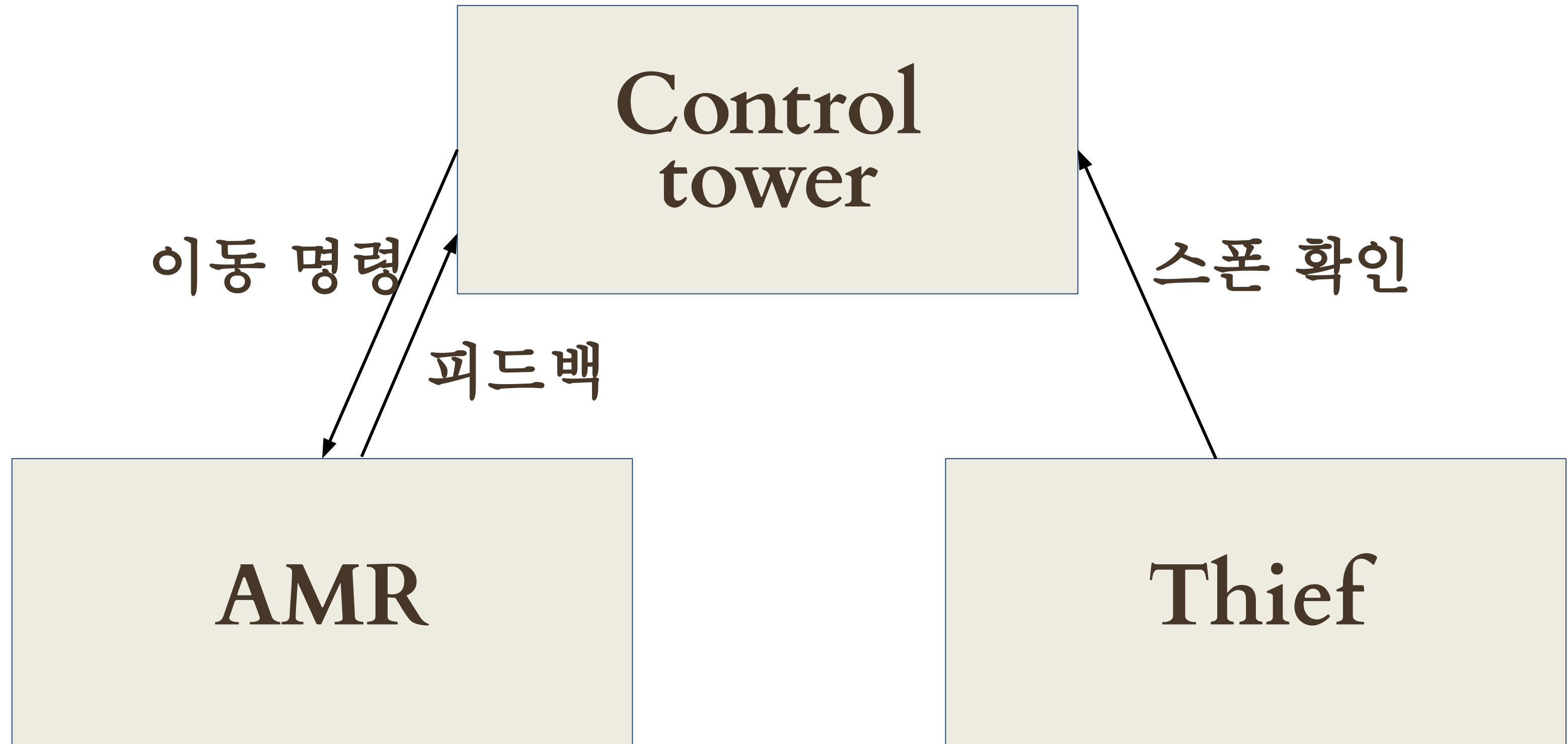
OverView



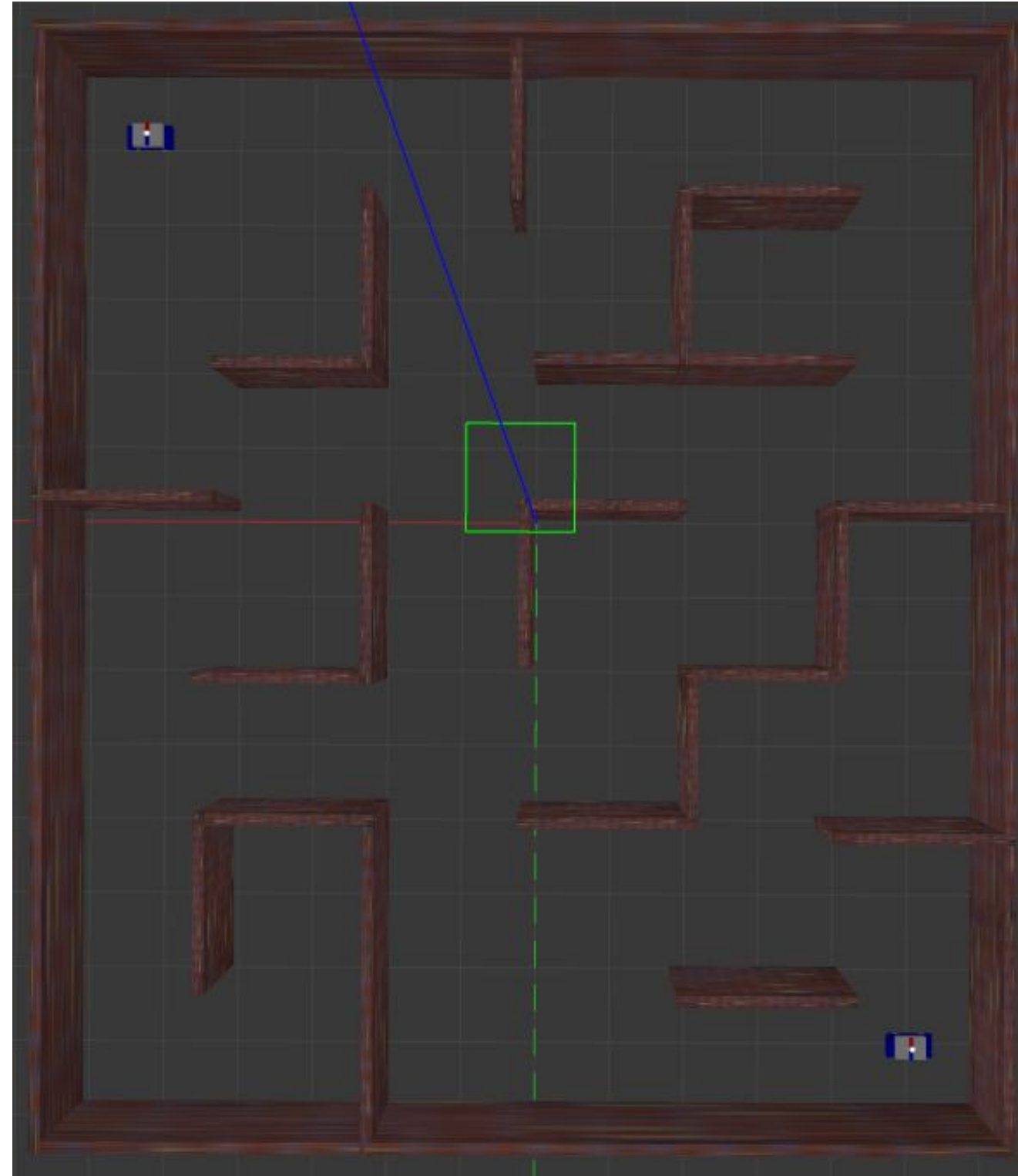
야간 경비 인원의 안전

야간 경비 인원 감축

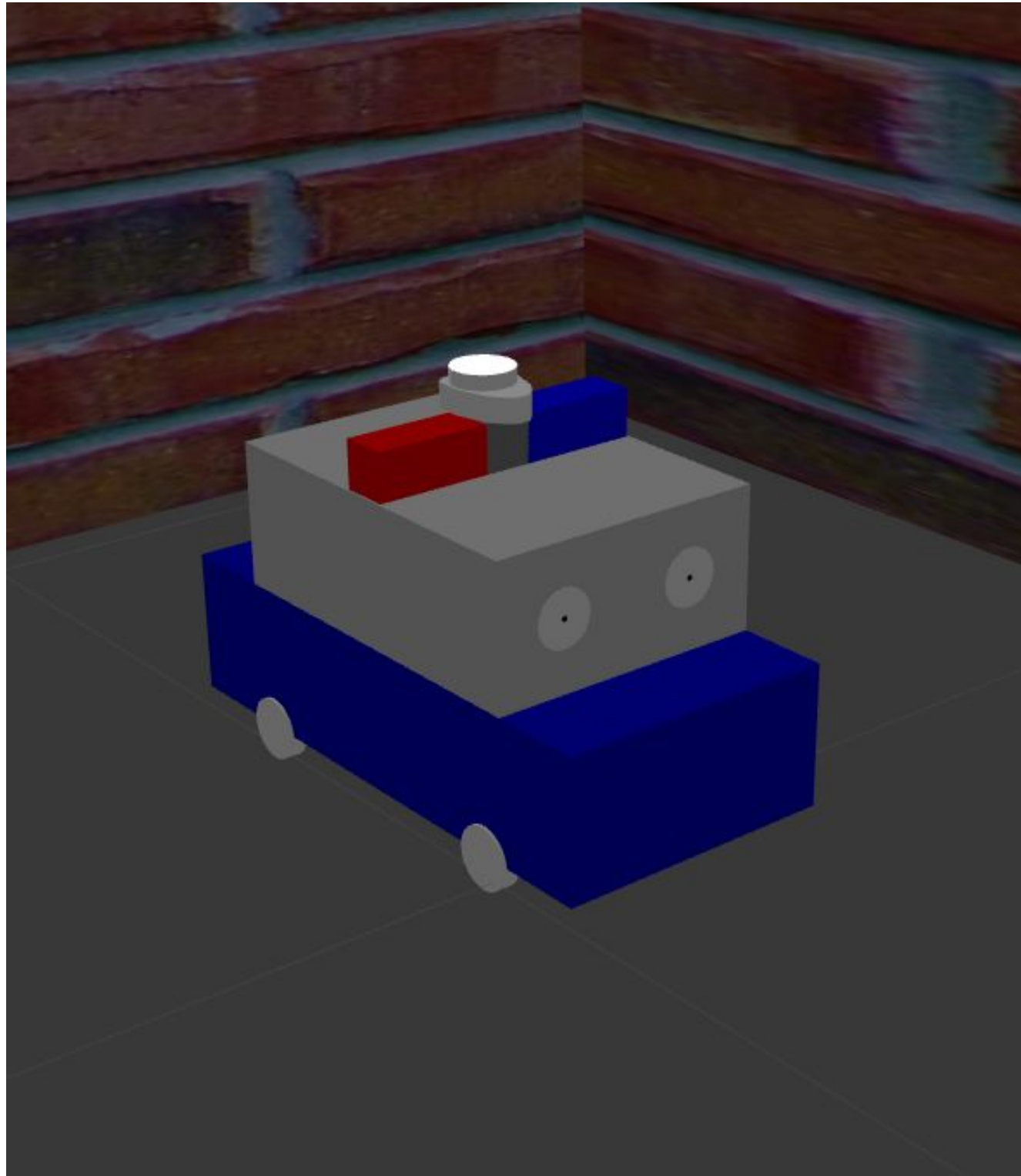
아키텍처



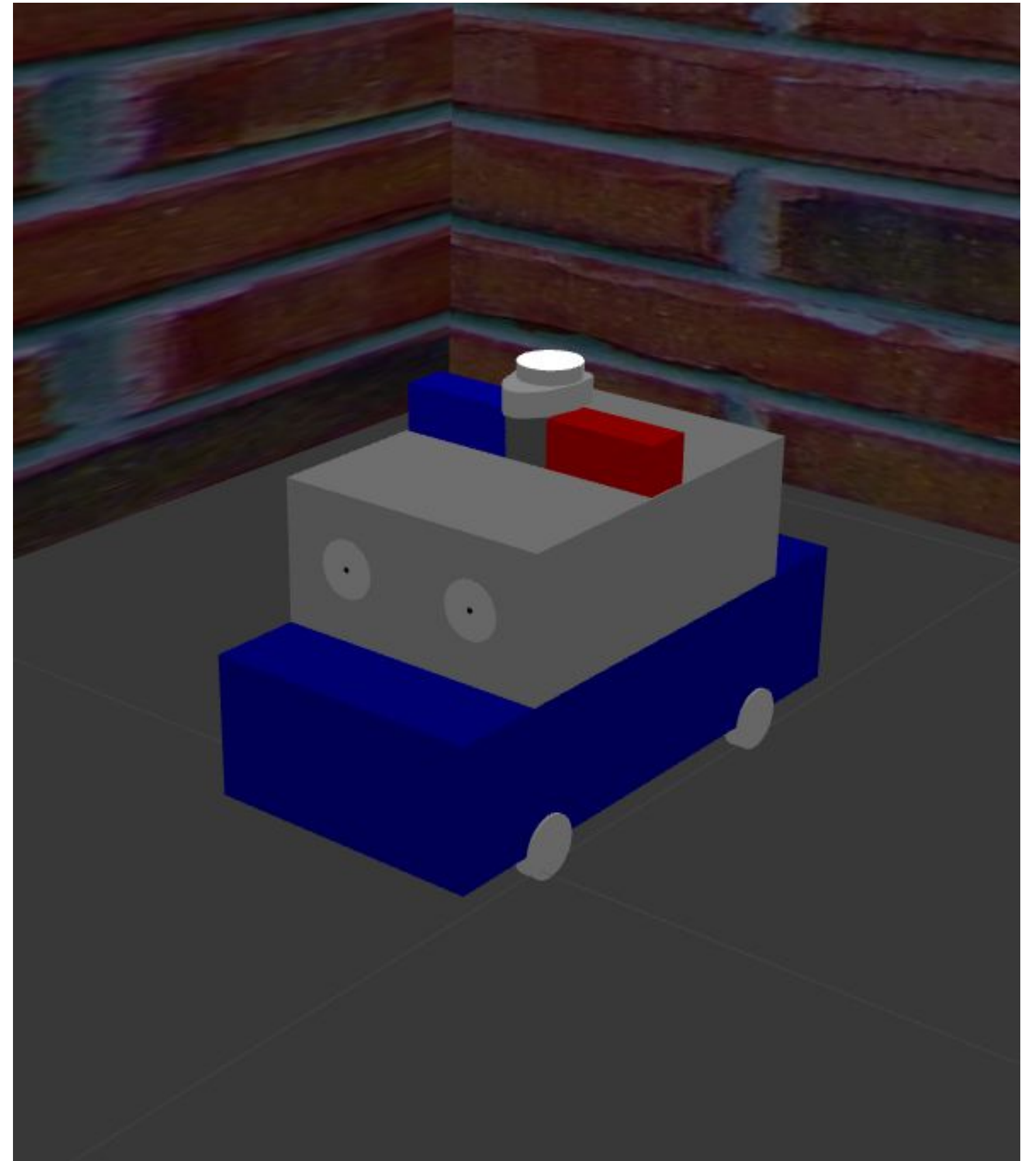
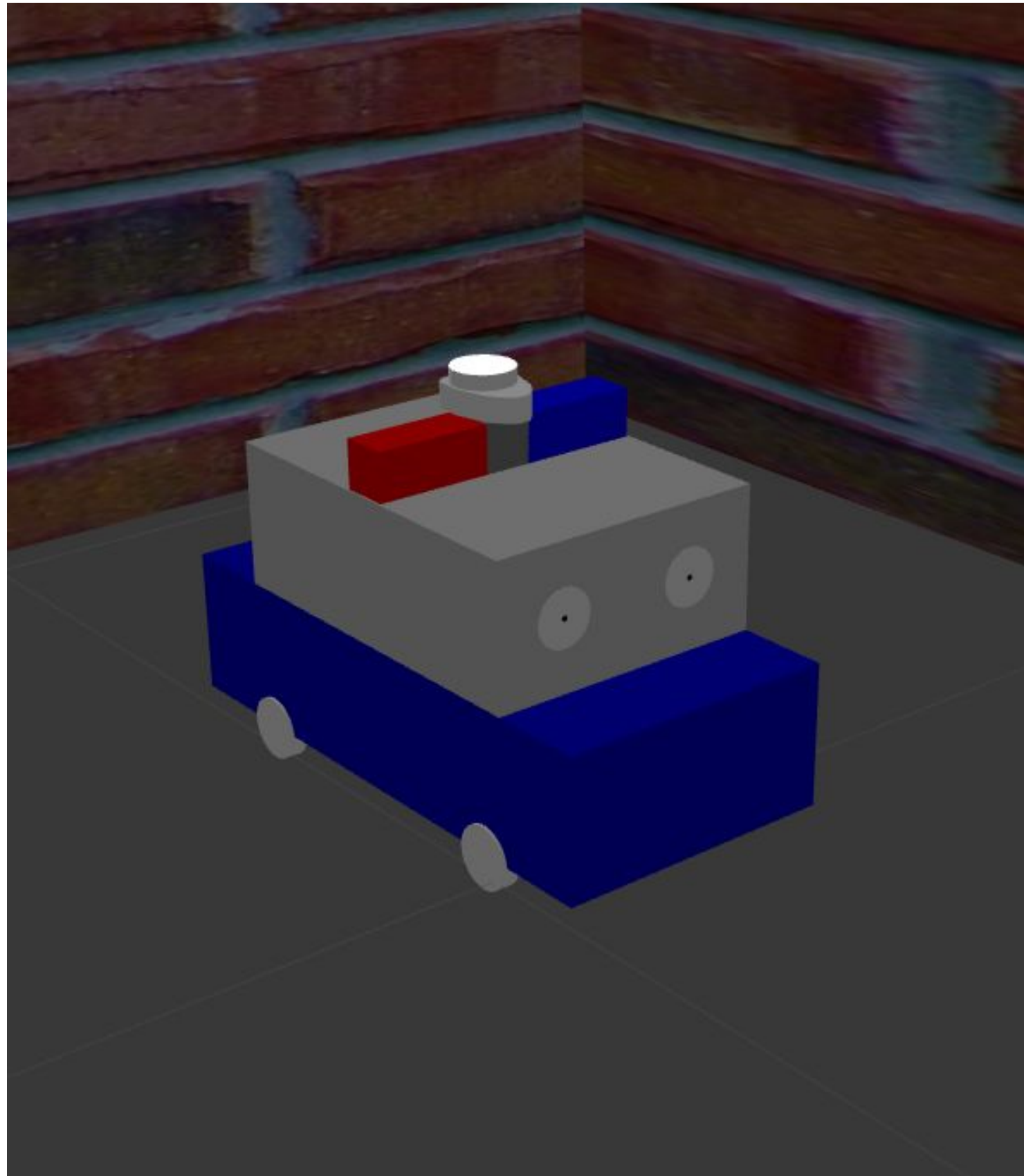
월드맵



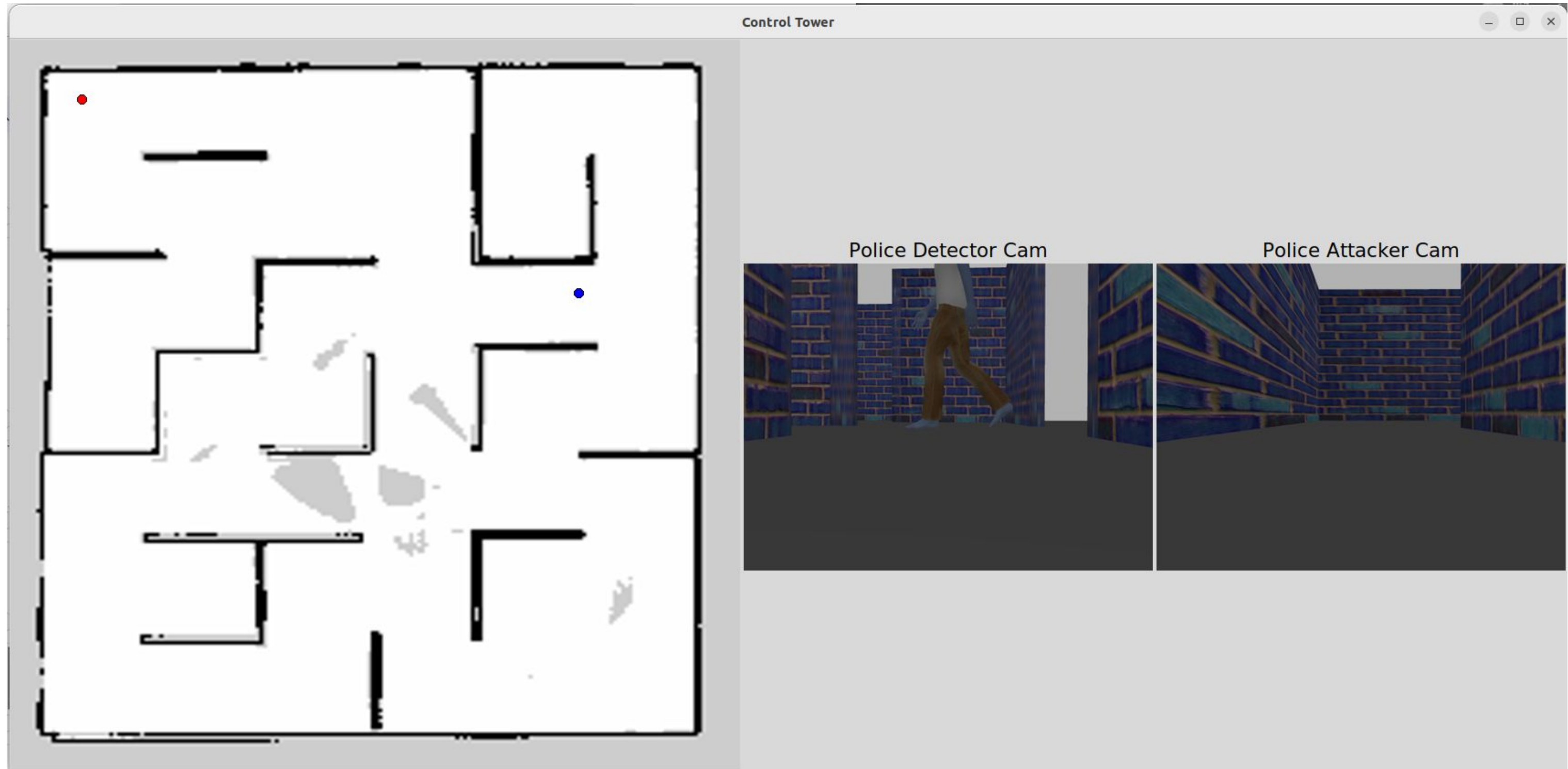
경찰과 도둑



경찰



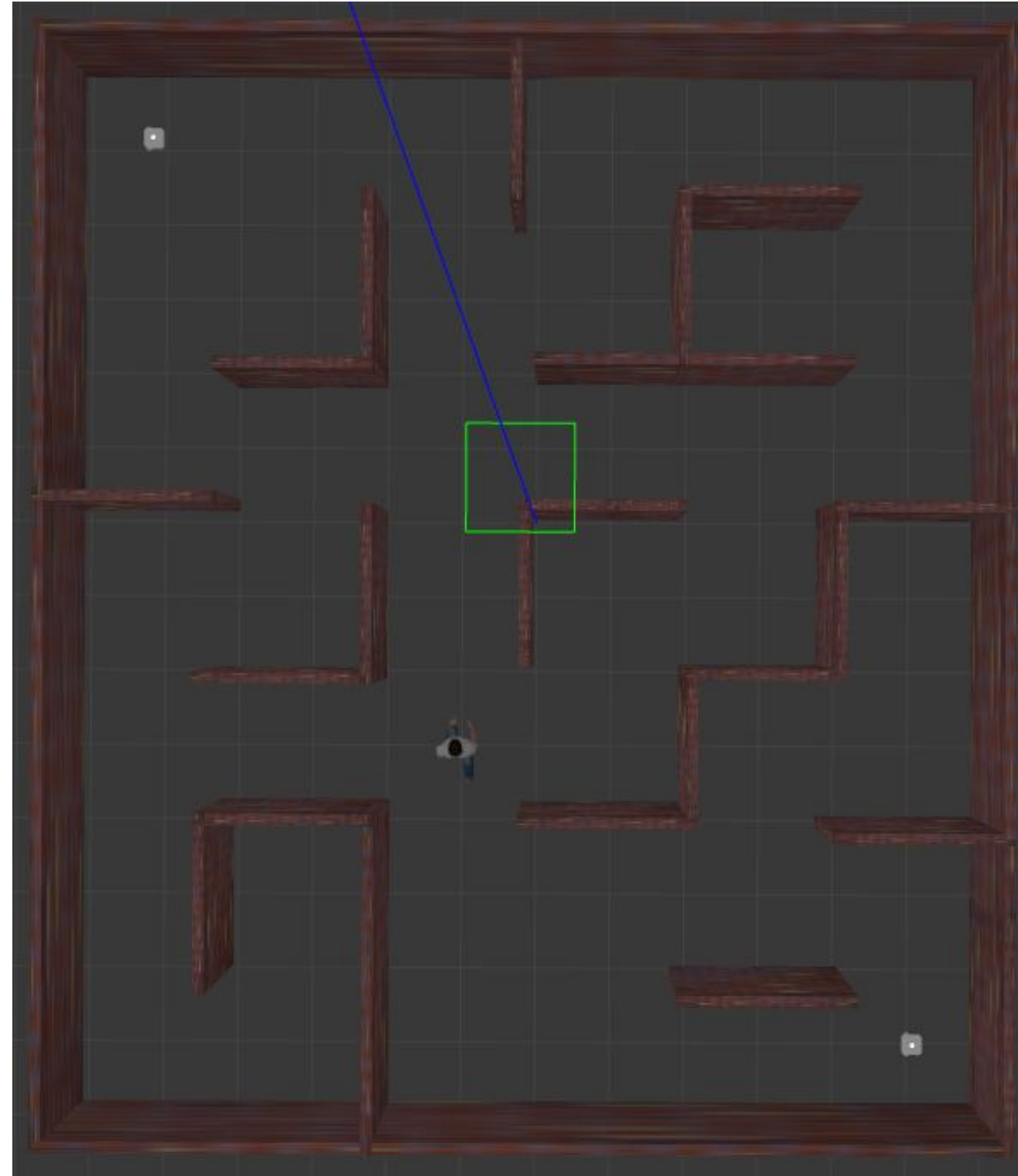
중앙관제 시스템



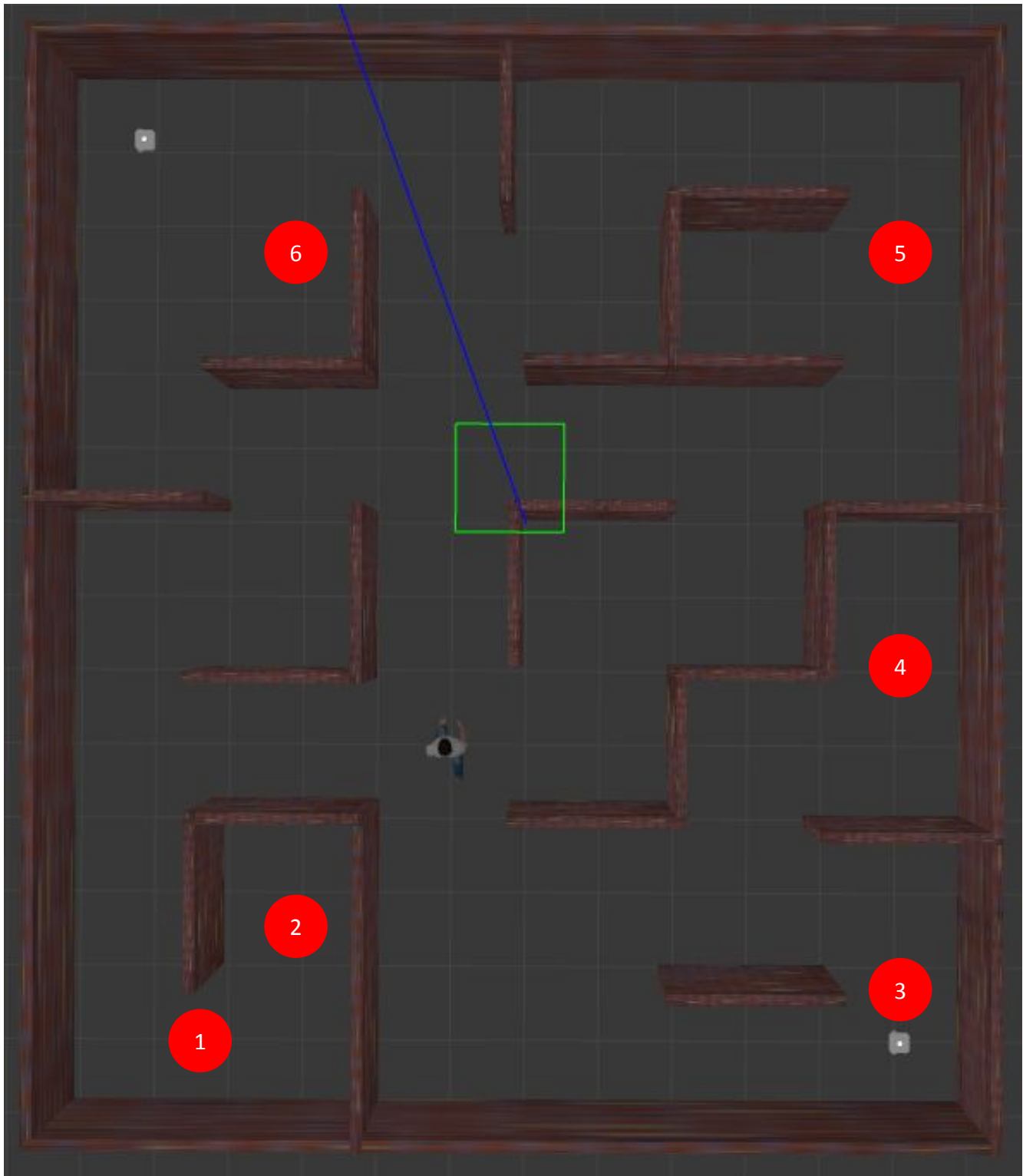
알고리즘

1. 도둑 랜덤 위치 스폰
2. 경찰차1(정찰) 수색
3. 경찰차2(폭격) 도둑 위치 이동
4. 도둑에게 무차별 폭격
5. 범인 검거 완료

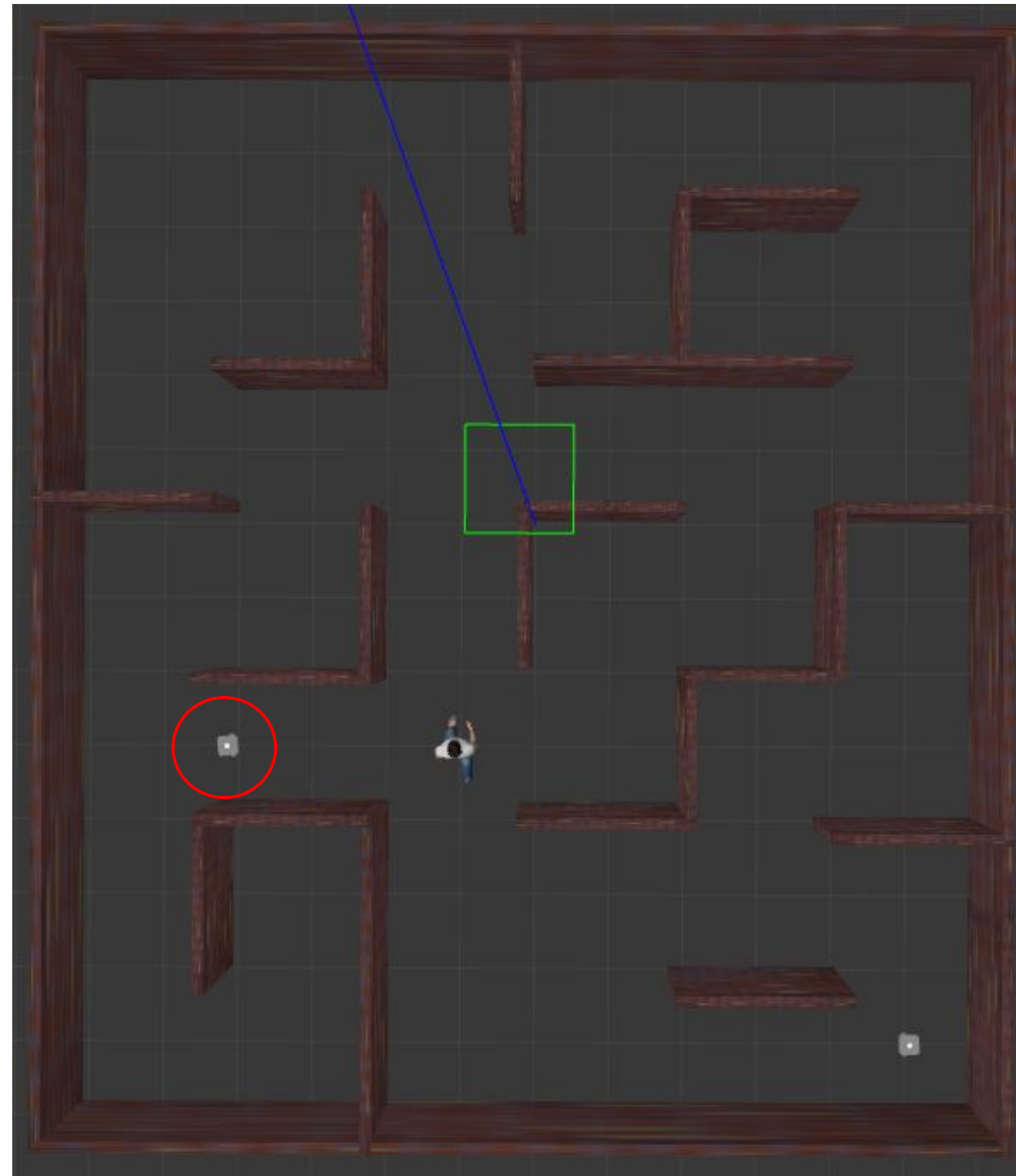
알고리즘



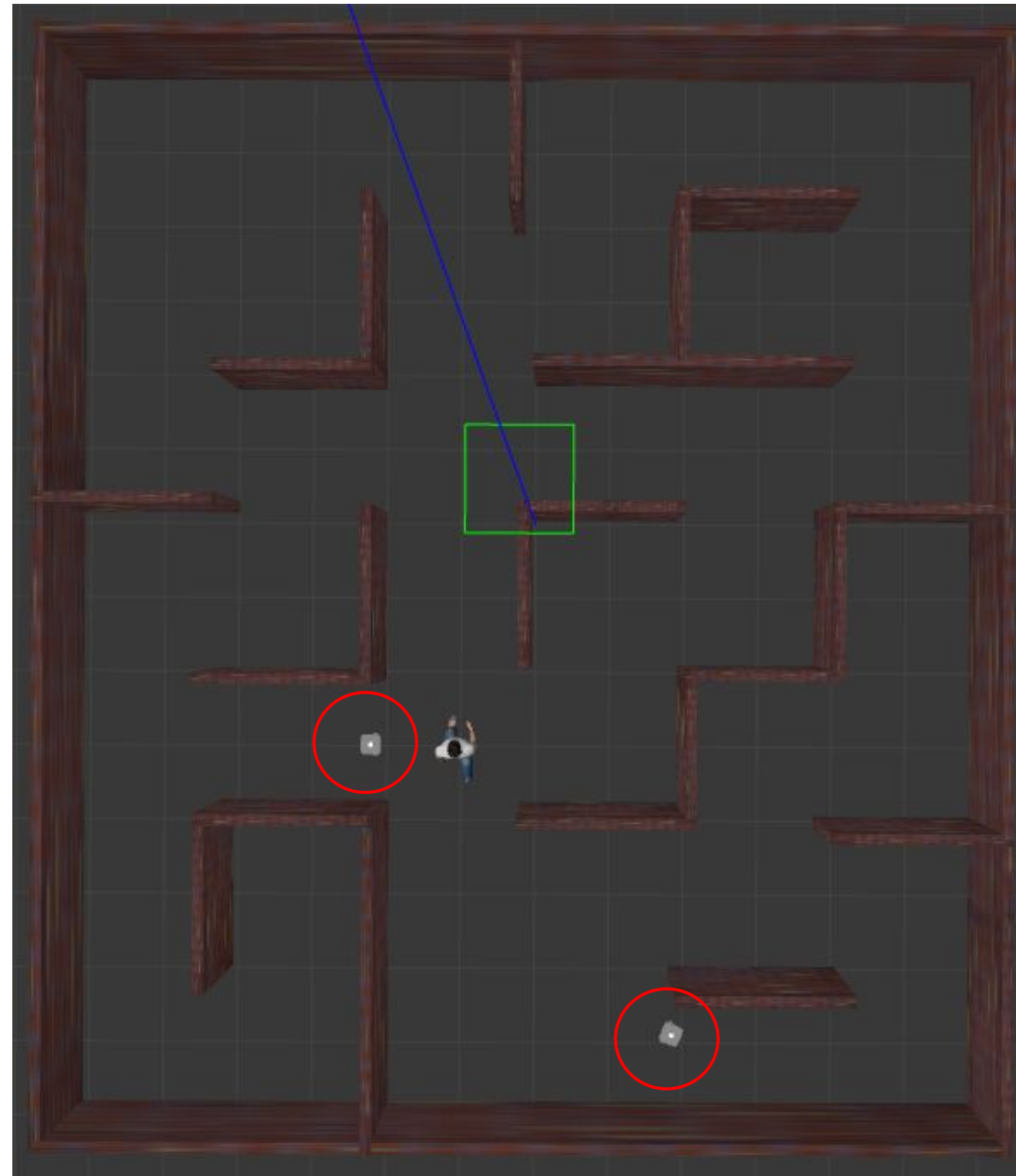
알고리즘



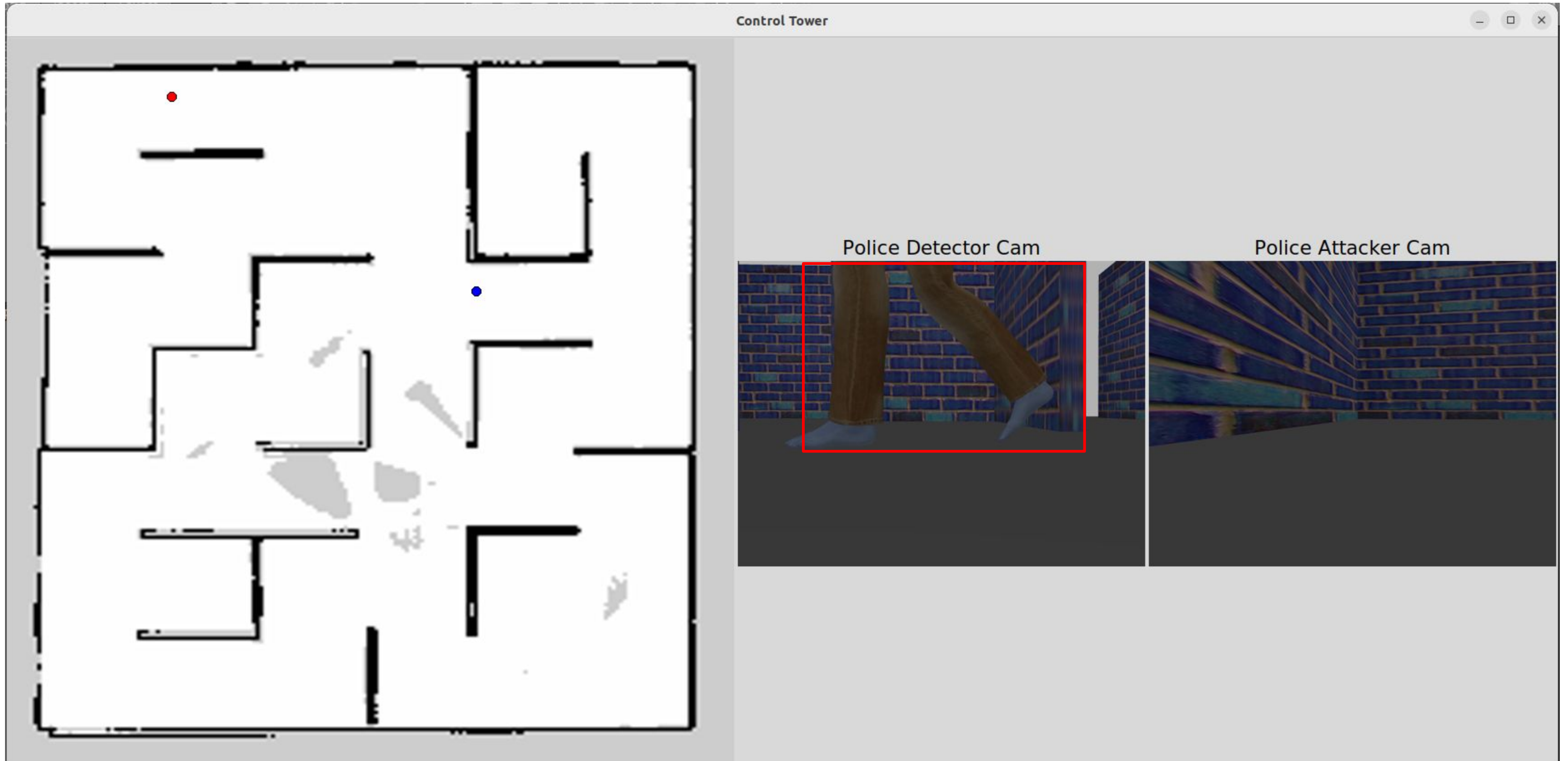
알고리즘



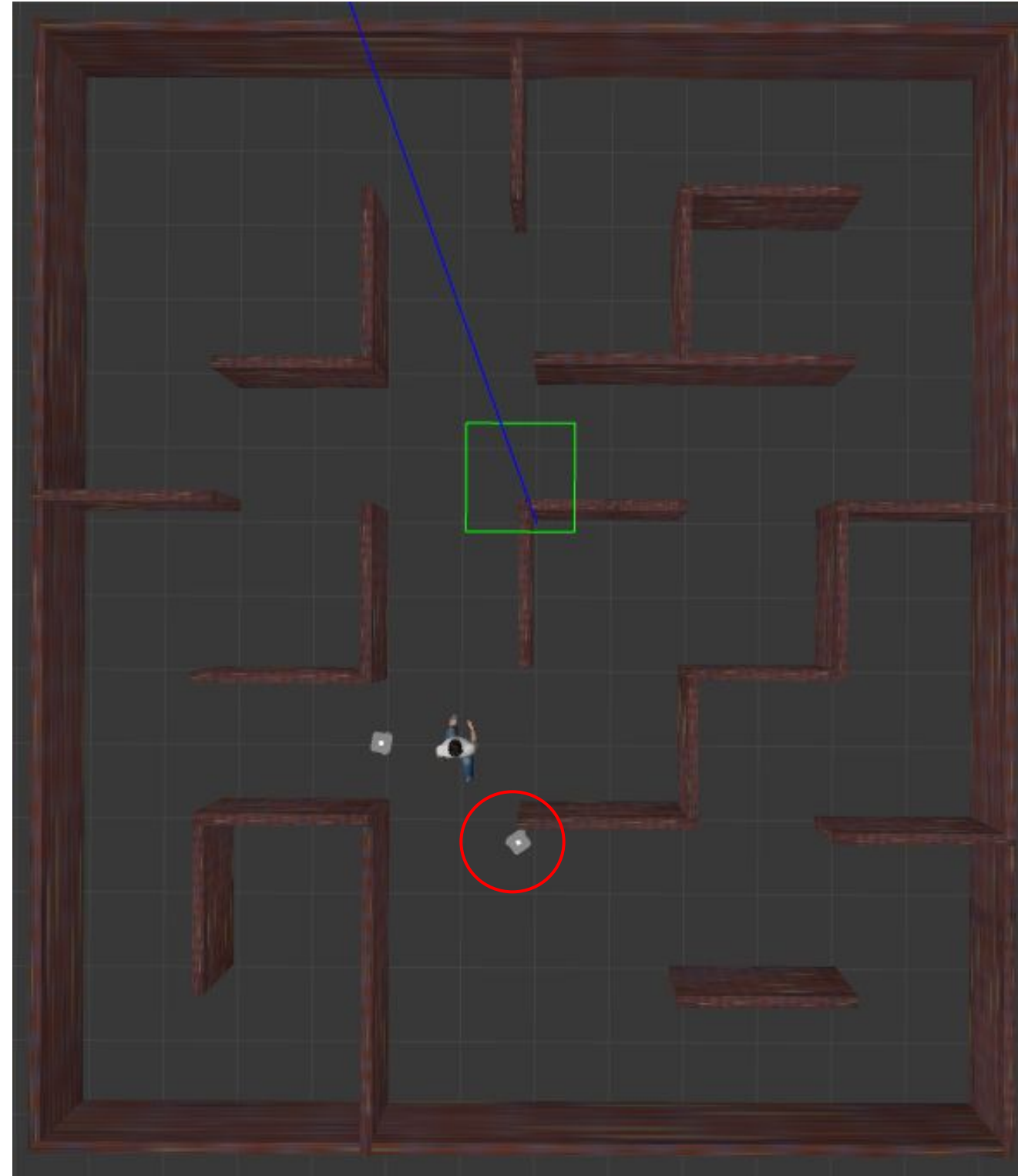
알고리즘



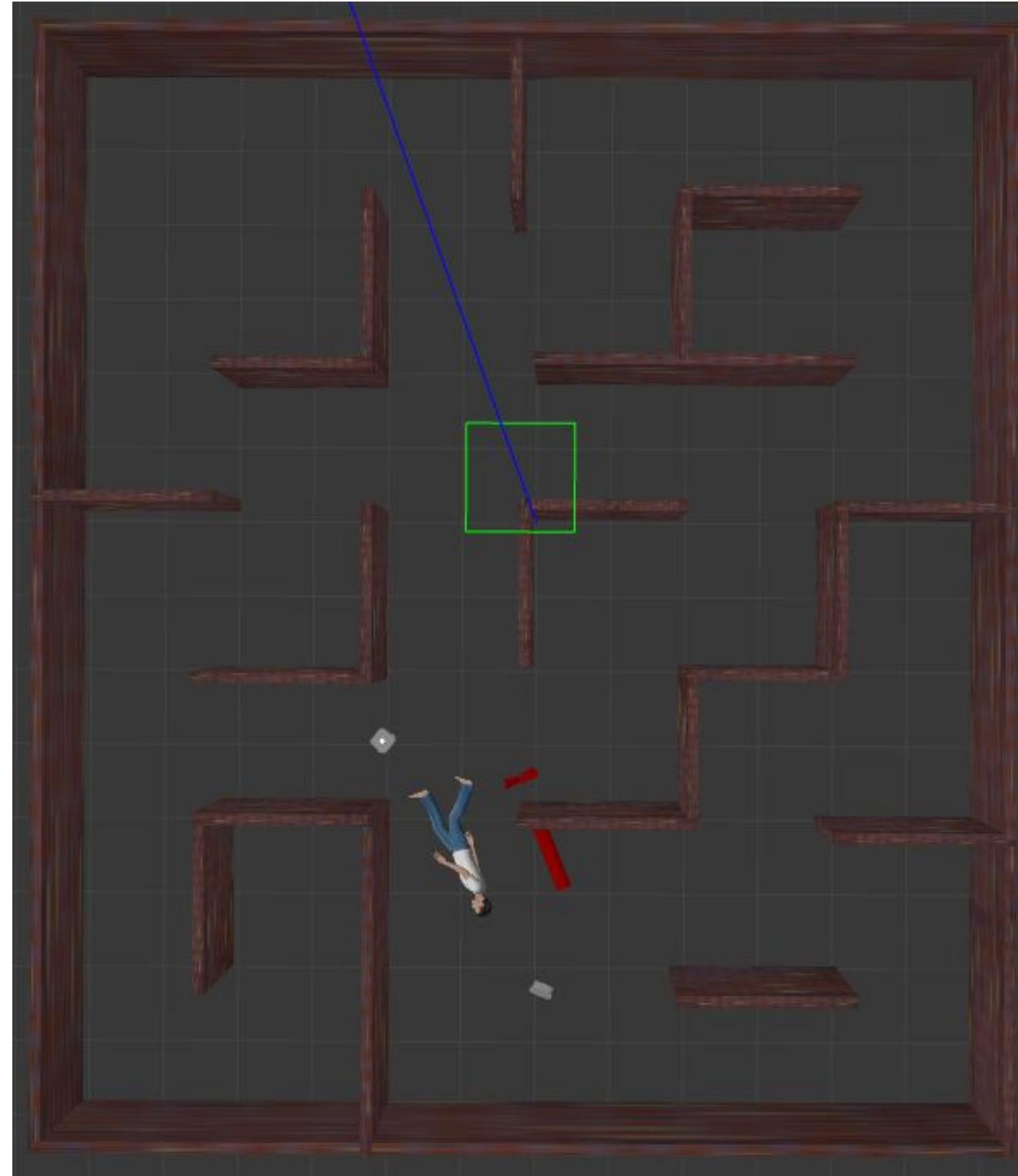
알고리즘



알고리즘



알고리즘



어려웠던 점

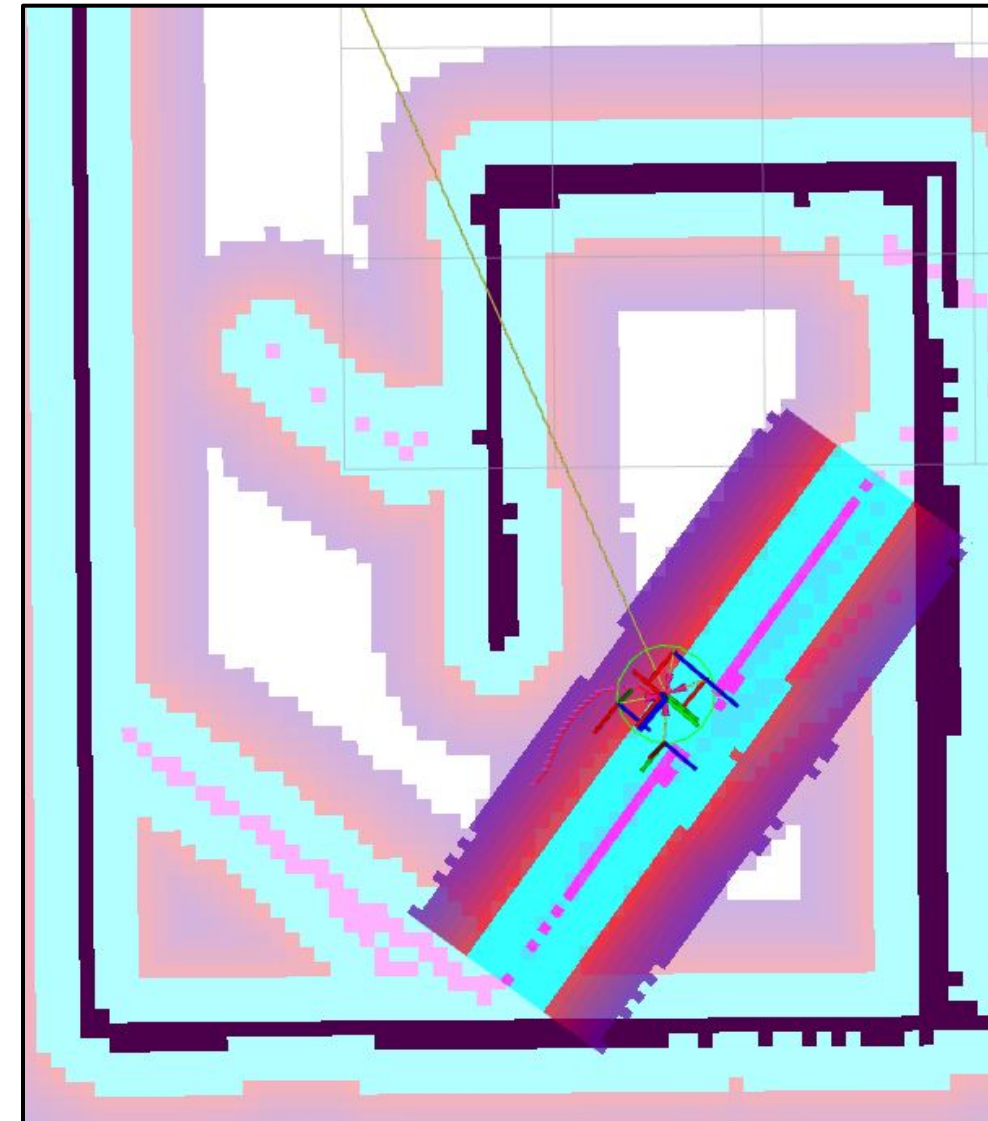
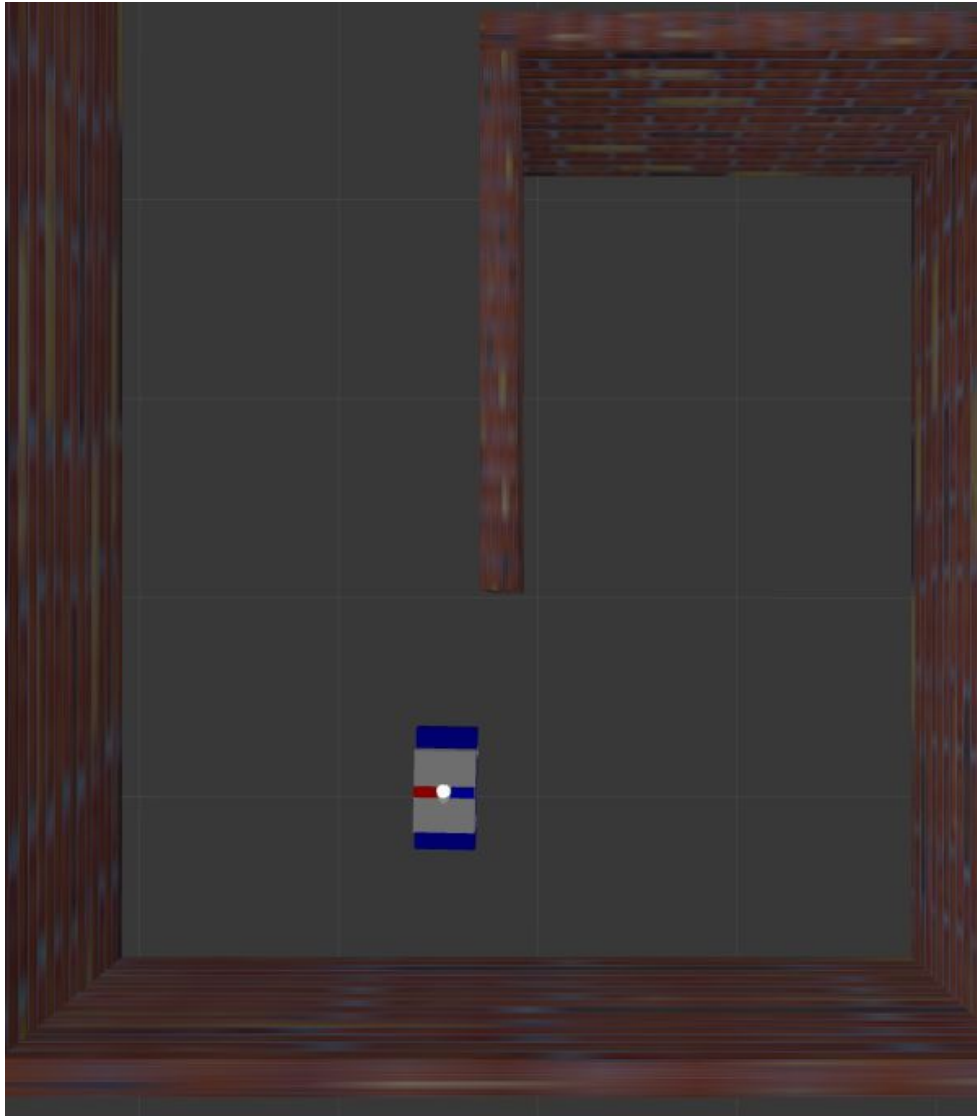
1. 시뮬레이션 상 이미지 토평 전달

지연

```
average rate: 0.858  
    min: 1.055s max: 2.175s std dev: 0.27149s window: 15  
average rate: 0.809  
    min: 1.055s max: 2.301s std dev: 0.38047s window: 16  
average rate: 0.813  
    min: 1.055s max: 2.301s std dev: 0.36990s window: 17  
average rate: 0.818  
    min: 1.055s max: 2.301s std dev: 0.36060s window: 18  
average rate: 0.821  
    min: 1.055s max: 2.301s std dev: 0.35164s window: 19  
average rate: 0.825  
    min: 1.055s max: 2.301s std dev: 0.34355s window: 20  
average rate: 0.828  
    min: 1.055s max: 2.301s std dev: 0.33598s window: 21  
average rate: 0.831  
    min: 1.055s max: 2.301s std dev: 0.32870s window: 22
```


어려웠던 점

2. sdf, urdf로 직접 모델링 시 최적의 관성값 찾기 및 라이다 비트맵 문제



아쉬웠던 점 : 어슬렁 거리는 도둑 구현



```
import random

def generate_launch_description():
    ld = LaunchDescription()

    #랜덤좌표지정
    spawn_position = [[-3.0, -3.0, 0.5],
                      [3.0, 5.0, 0.5],
                      [-5.0, 1.0, 0.5]]

    random_spawn = random.choice(spawn_position)

    #좌표type변환
    thief_x = str(random_spawn[0])
    thief_y = str(random_spawn[1])

    # Names and poses of the robots
    robots = [
        {'name': 'tb1', 'x_pose': '5.234215', 'y_pose': '-5.155386', 'z_pose': '0.01'},
        {'name': 'tb2', 'x_pose': '-5.0084105', 'y_pose': '7.001489', 'z_pose': '0.01'},
        {'name': 'tb3', 'x_pose': thief_x, 'y_pose': thief_y, 'z_pose': '0.01'},
        {'name': 'tb4', 'x_pose': '1.5', 'y_pose': '0.5', 'z_pose': '0.01'},
        # ...
        # ...
    ]

    #도둑 모델 경로 설정, tb3 이름 변경 부분, arguments 의 file에 적용 #####
    if robot['name'] == 'tb3':
        model_file = os.path.join(os.getenv('HOME'), 'thief_model', 'model.sdf')
    else:
        # 경로, 폴더 이름만 순서대로
        model_file = os.path.join(turtlebot3_multi_robot, 'models', 'turtlebot3_' + TURTLEBOT3_MODEL, 'model.sdf')

    # Create spawn call
    spawn_turtlebot3_burger = Node(
        package='gazebo_ros',
        executable='spawn_entity.py',
        arguments=[
            '-file', model_file, ##
            '-entity', robot['name'],
            '-robot_namespace', namespace,
            '-x', robot['x_pose'], '-y', robot['y_pose'],
            '-z', '0.01', '-Y', '0.0',
            '-unpause',
        ],
        output='screen',
    )
```

프로젝트 시연



Thank You

D - 2