

---

# 객체지향개발방법론 (OOAD)

로봇 청소기(RVC) OOI

202211291 김호경

202111282 노을영

202111284 문재현

---

# Agenda

[https://github.com/davidpro08/2026\\_1\\_OOAD](https://github.com/davidpro08/2026_1_OOAD)

- 수정 사항
- Coding
  - Pair Programming
  - Unit Test
  - Code Review
- System Test
  - Simulator
  - System Cases
  - Results
- Static Code Analysis
  - Code Smell
- Trouble Shooting

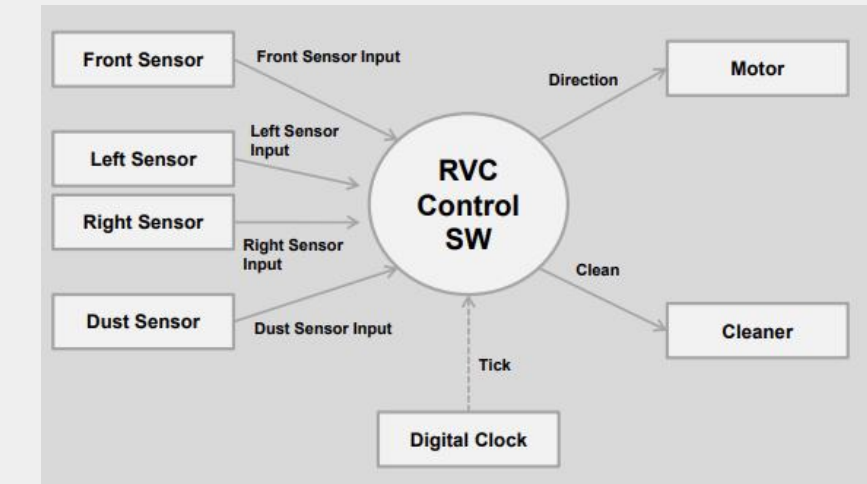
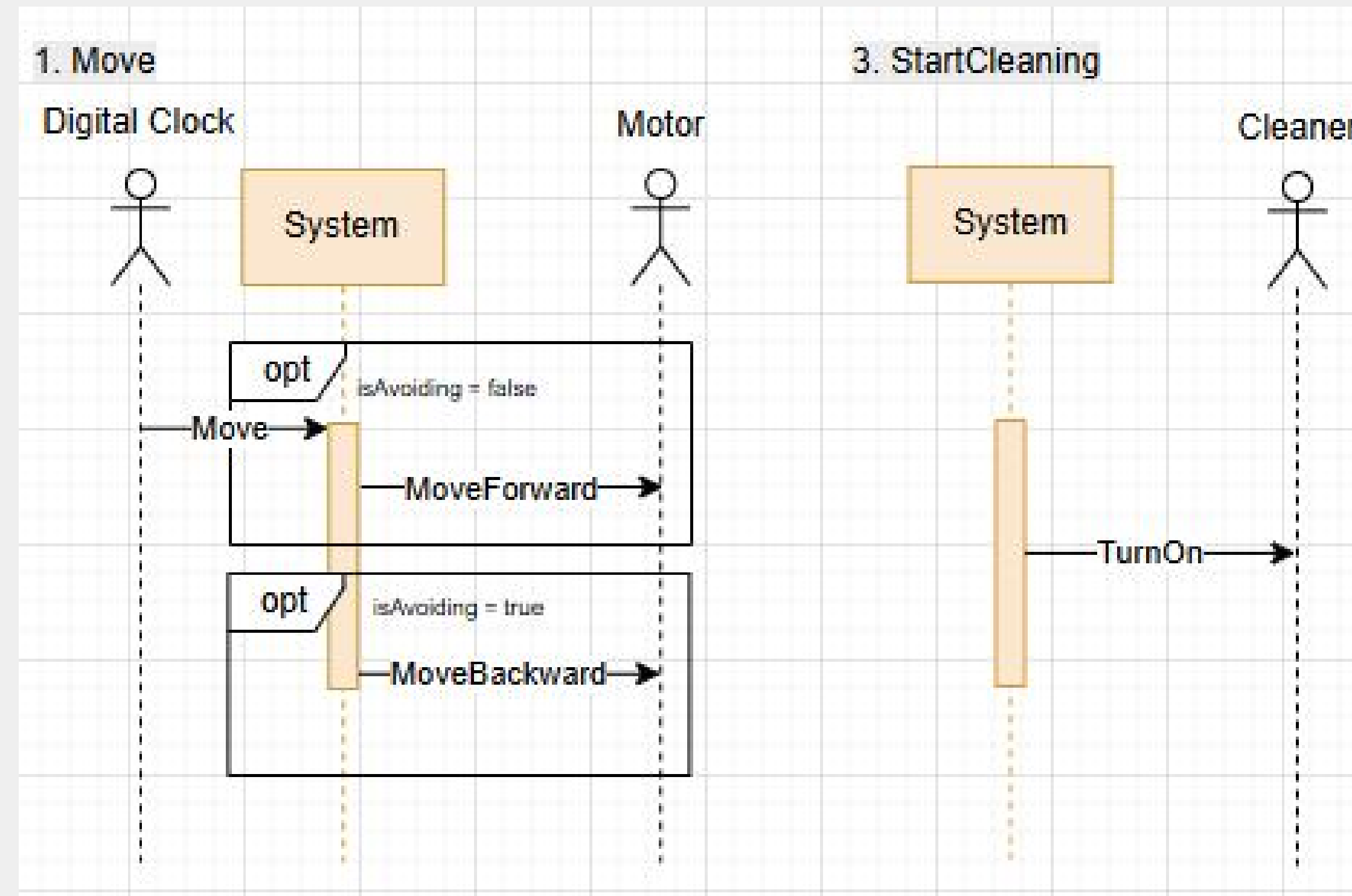
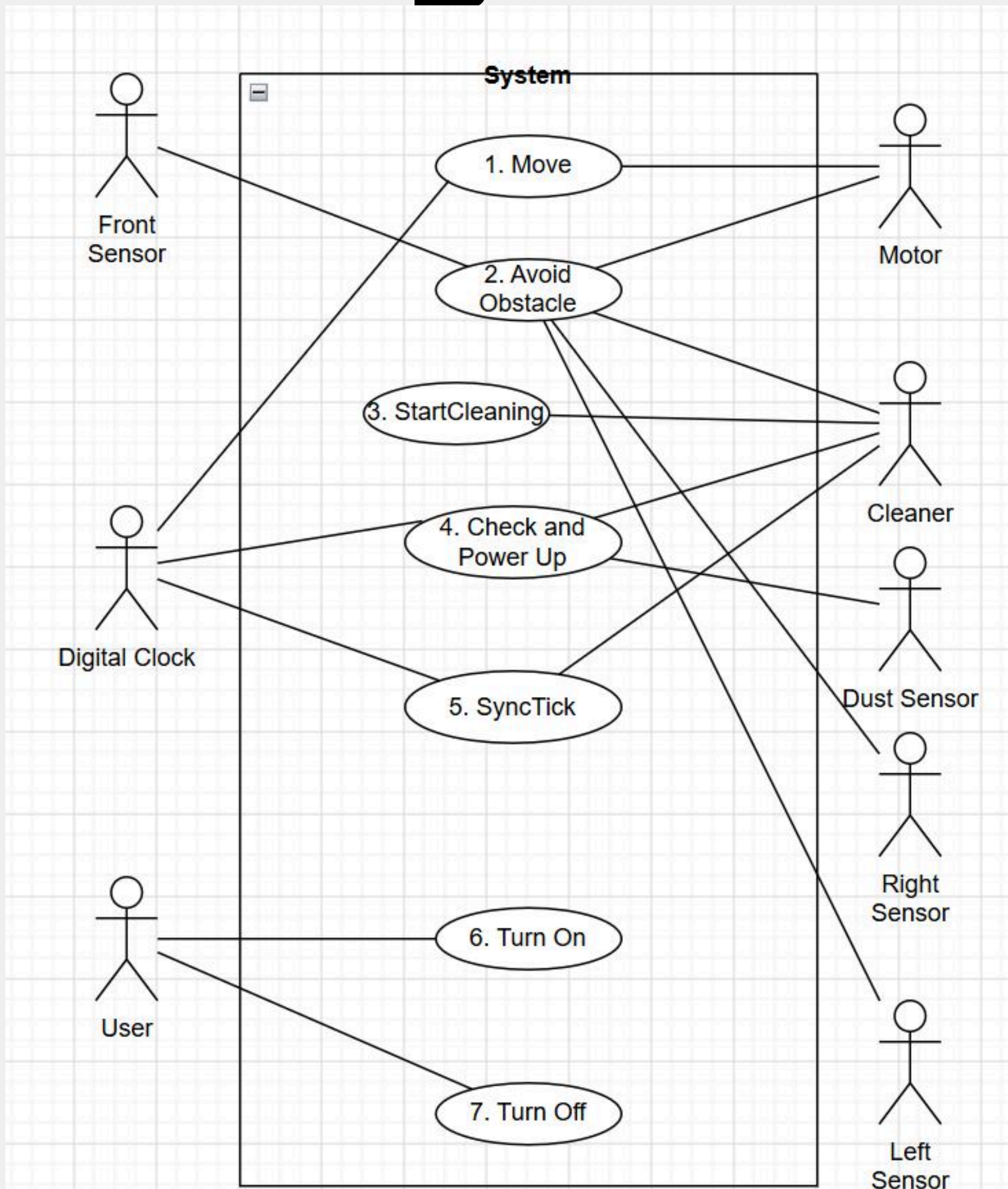
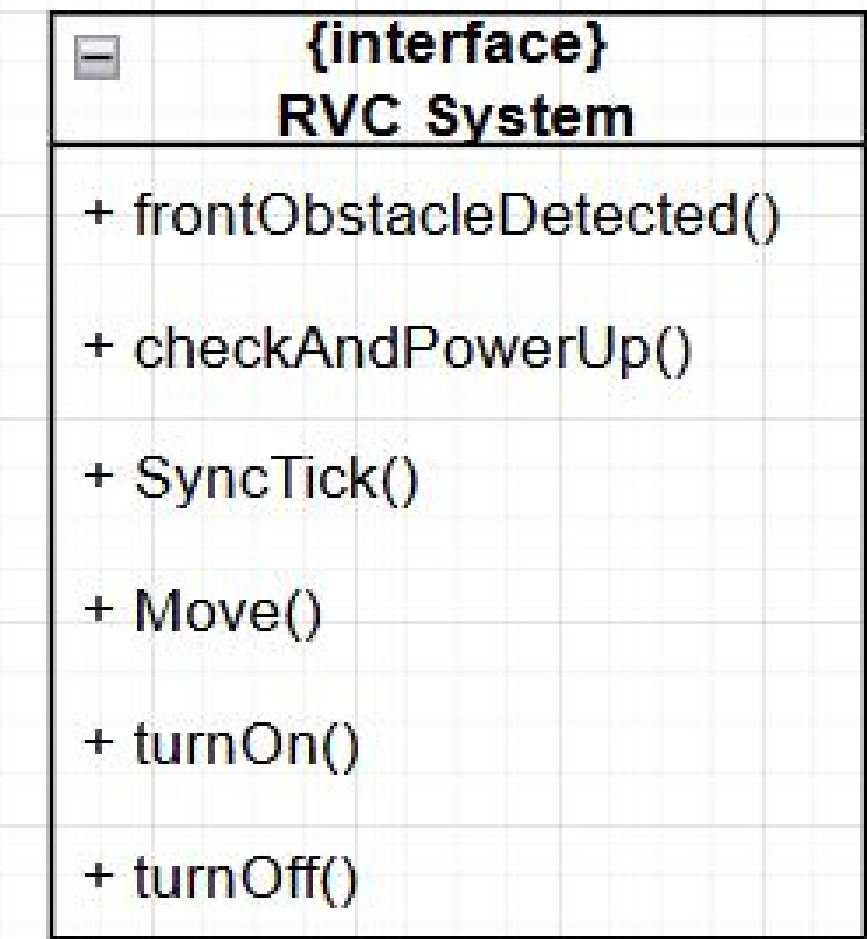
---

# 수정 사항

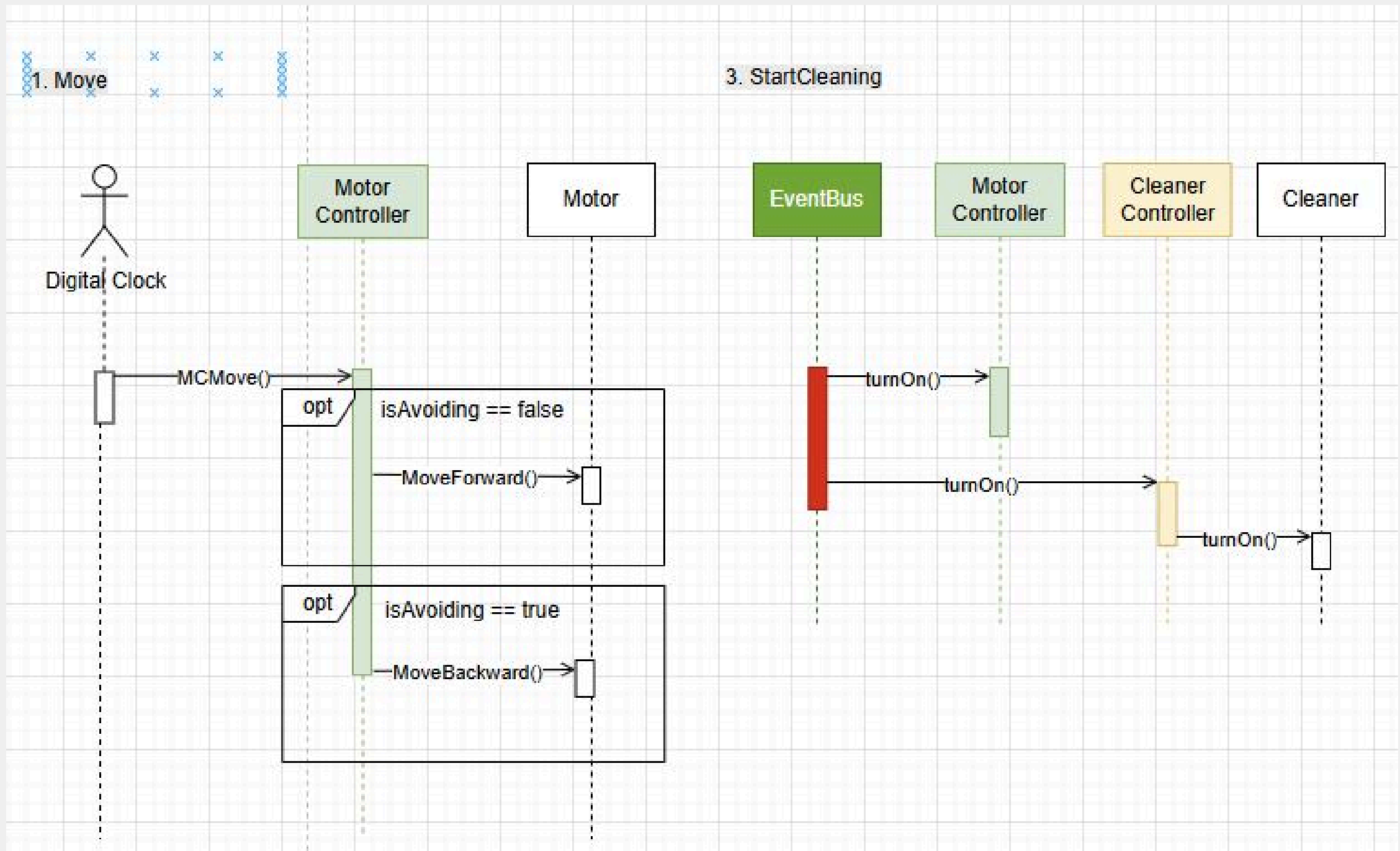
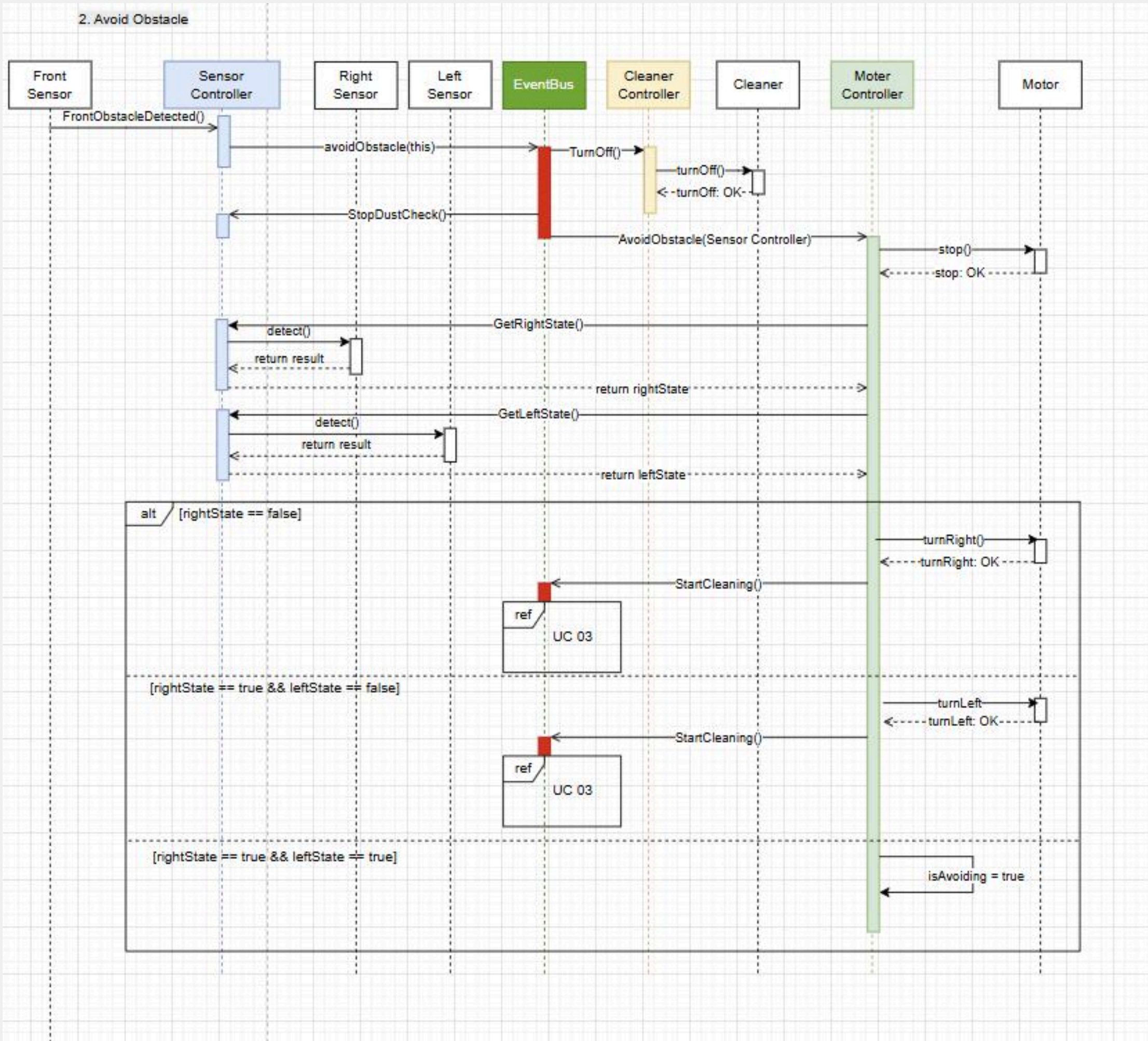
---

# UseCase

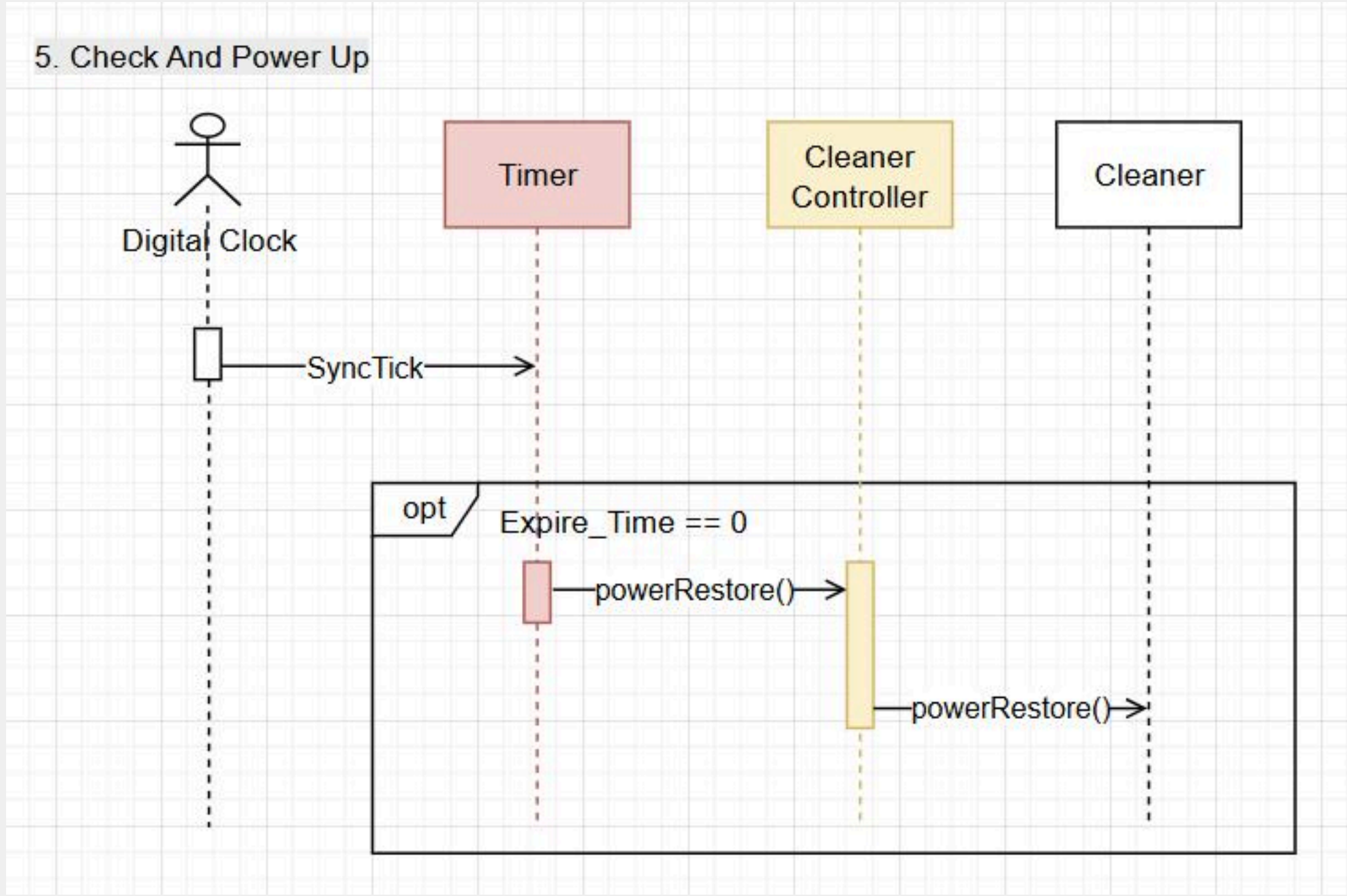
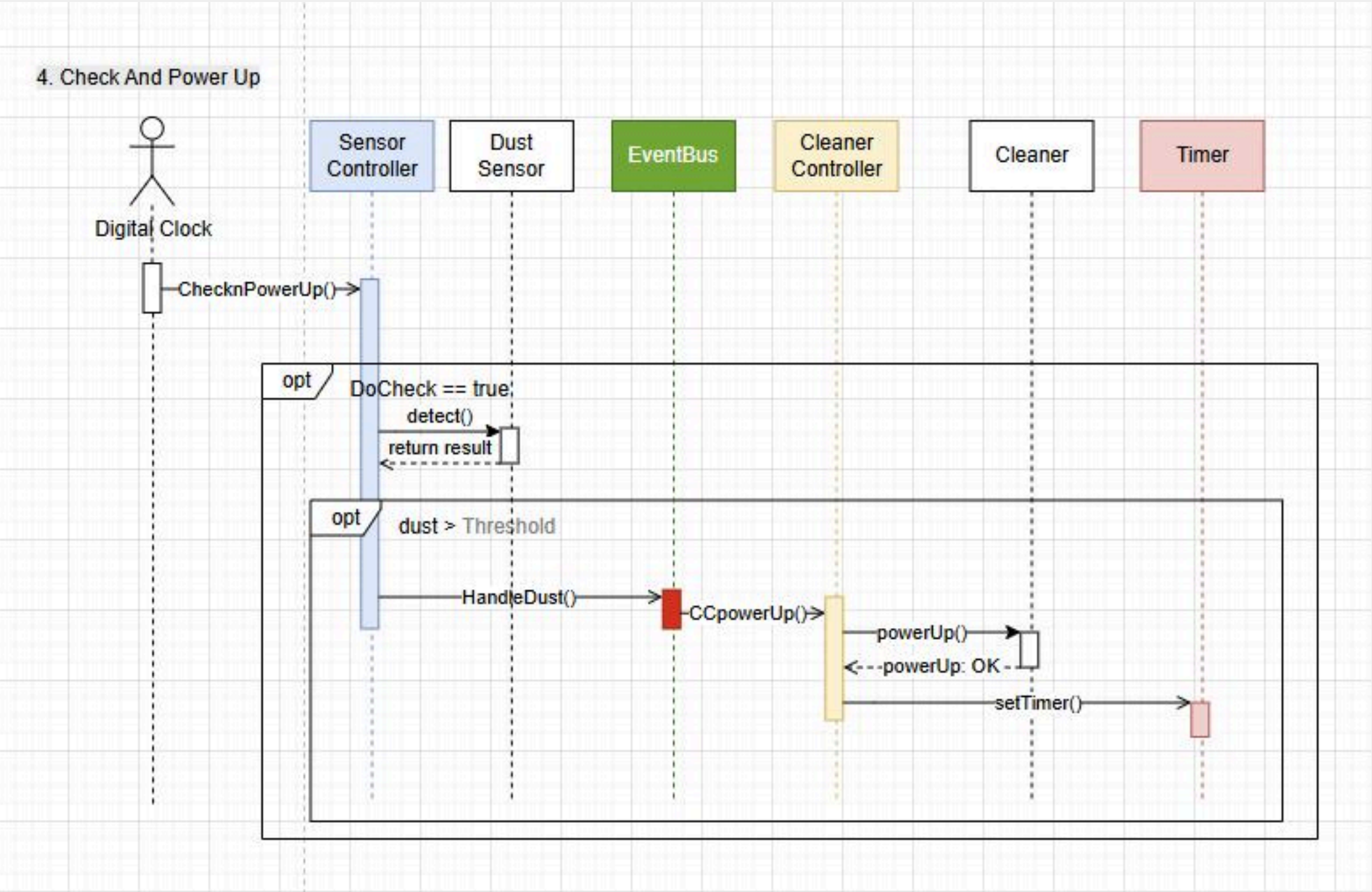
# 및 System operation 추가



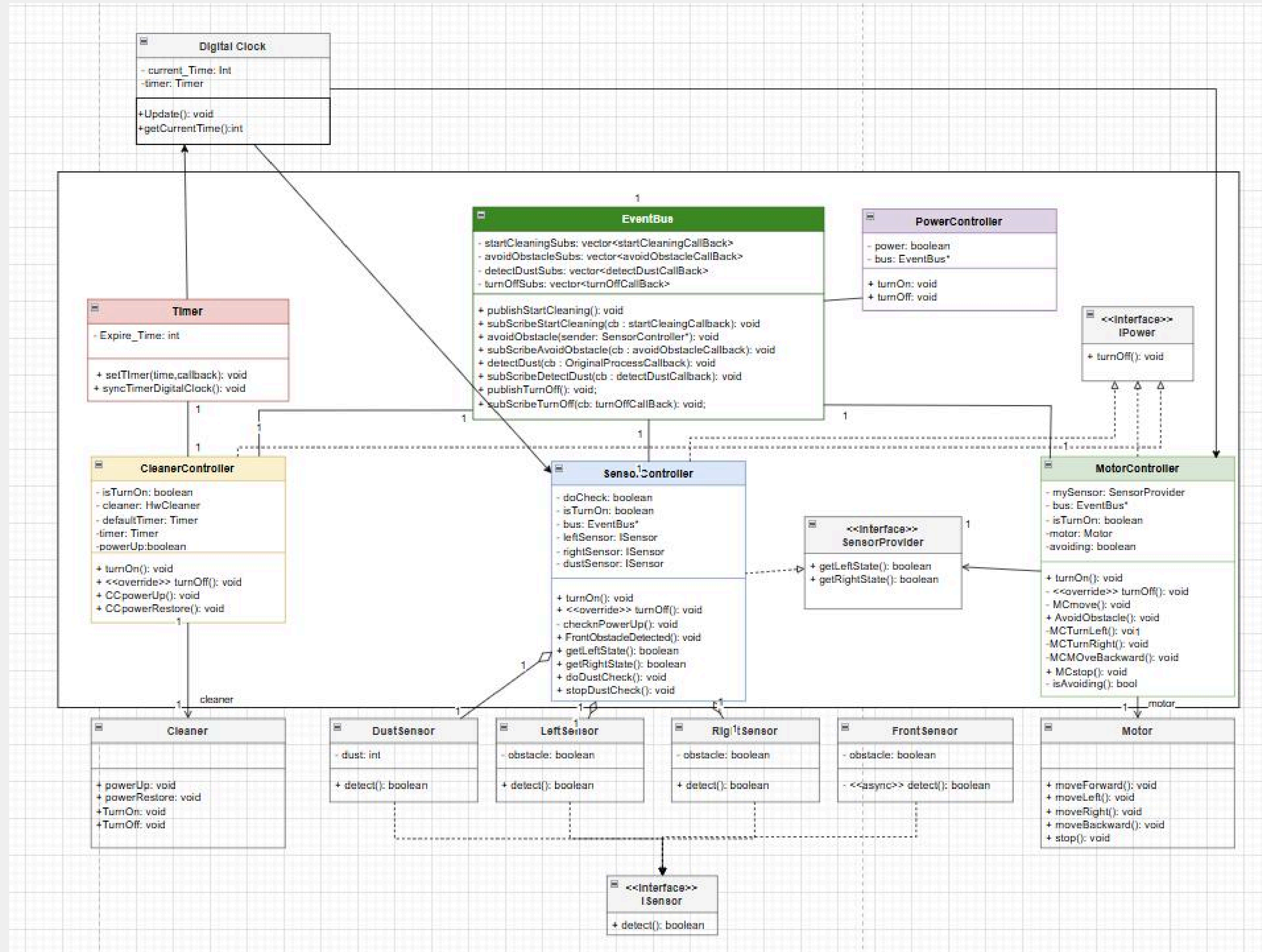
# Sequence Diagram



# Sequence Diagram



# Class Diagram

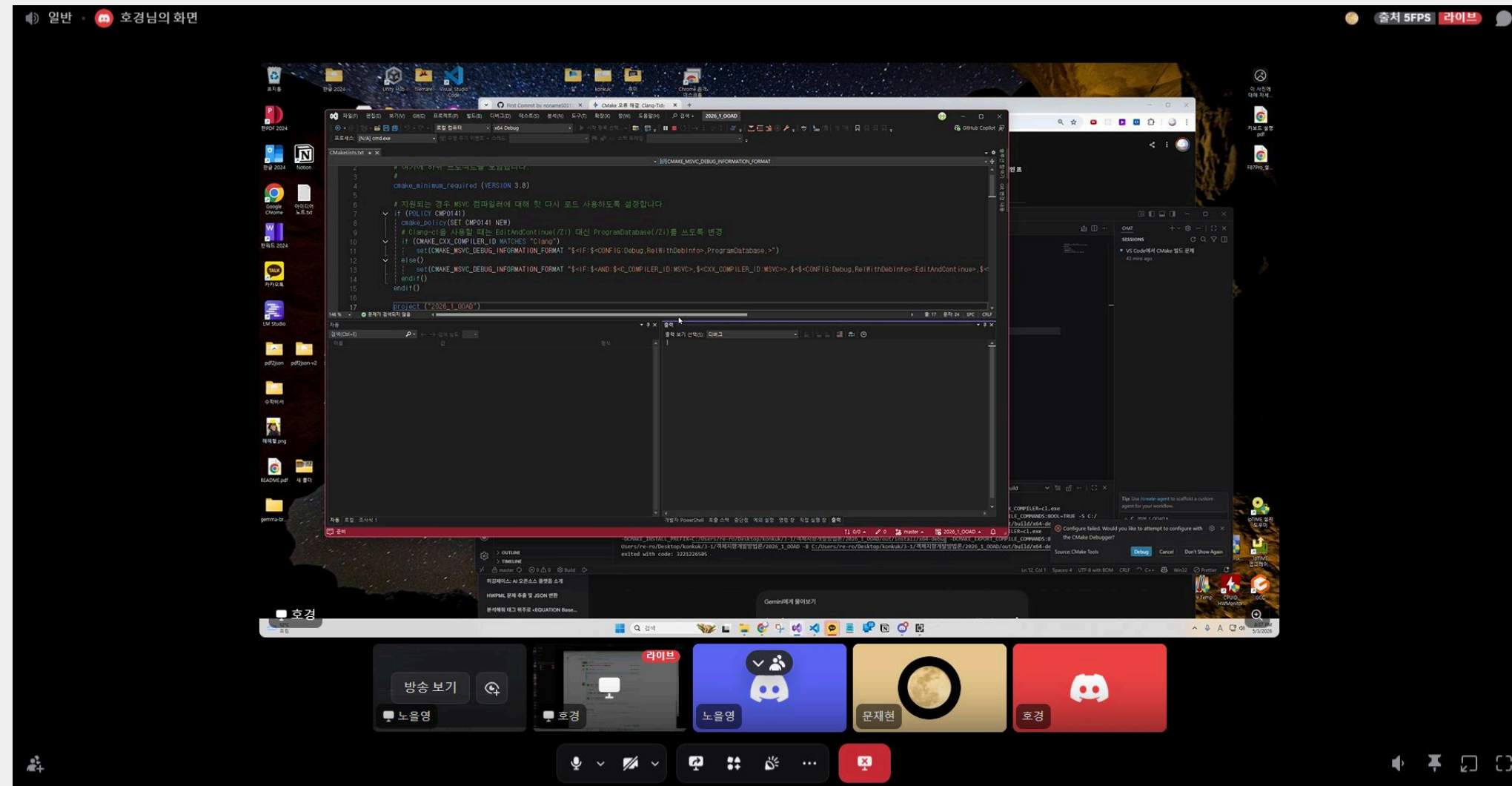


---

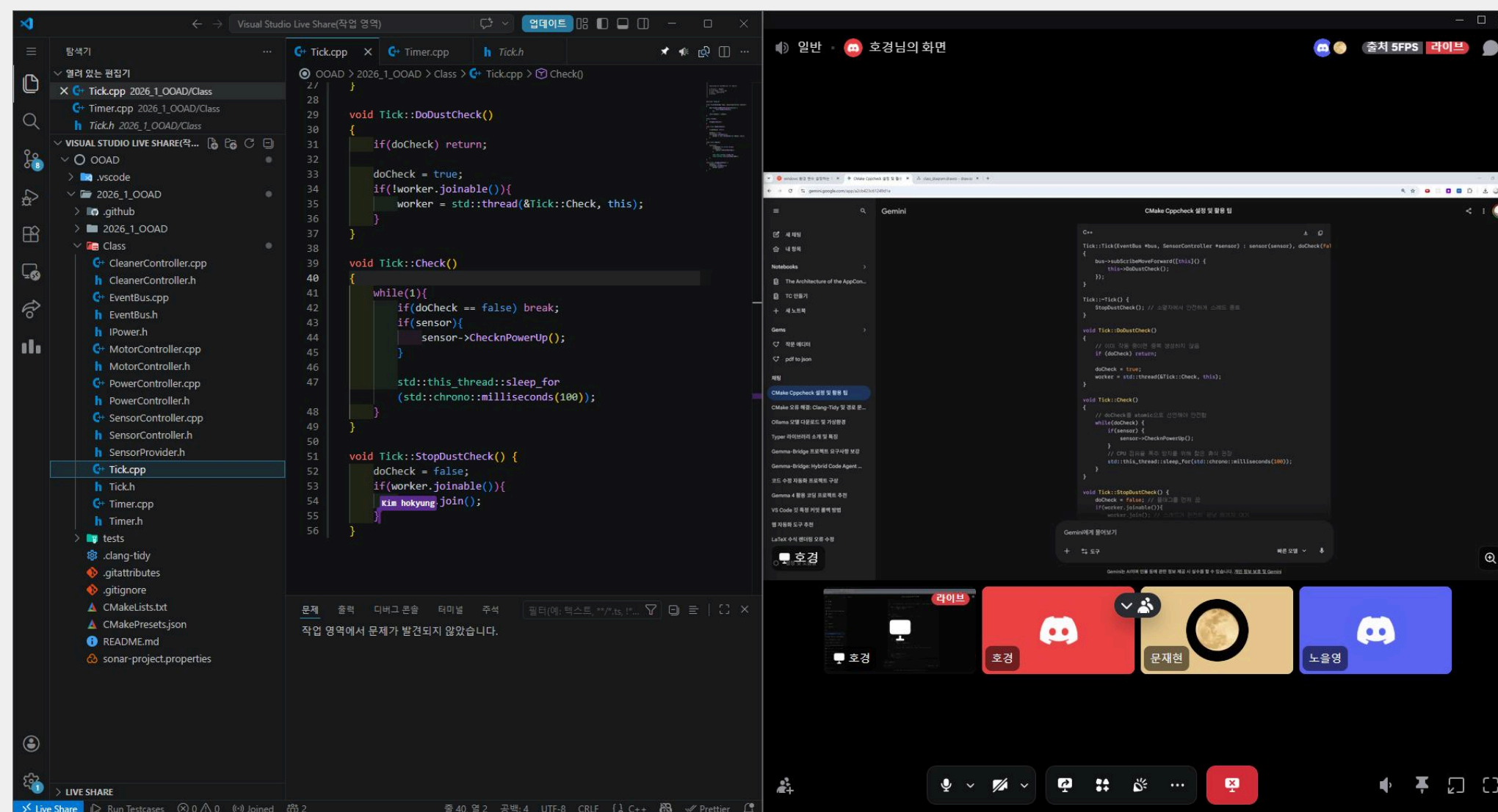
# CODING

---

# Pair Programming

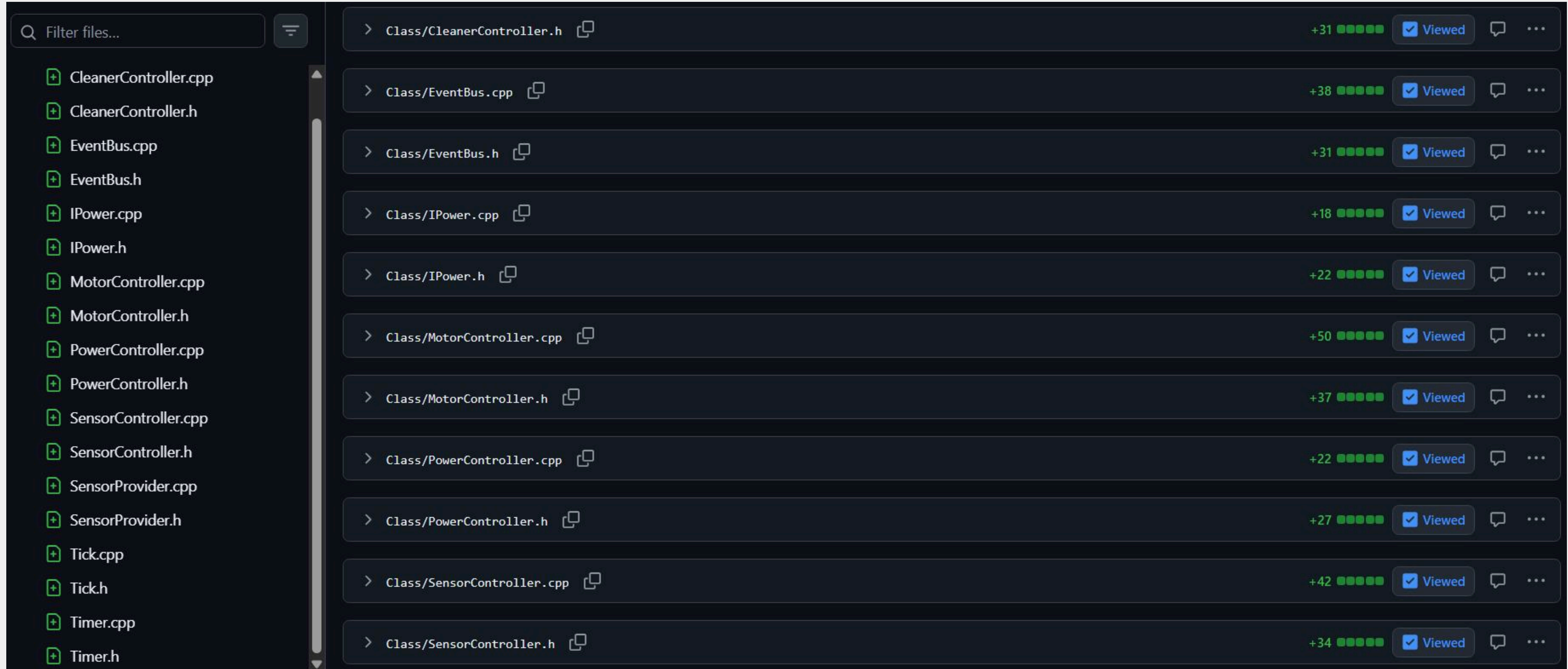


- 2인 1조가 원칙이지만, 팀원이 3명이어서 3명이서 같이 페어 프로그래밍
  - Discord 화면 공유
  - 한 명은 문서를 아예 보지 않음
  - 나머지 두명이 문서를 보고 로직 설명을 해주면 문서를 보지 않는 한 명이 코드를 짤
  - 잘못 구현된 부분을 바로바로 고치기
- Discord 화면 공유로는 부족한 점이 있다 생각해서, VSCode Live Share 사용



- 한 명이 코드를 짜는 건 동일
- 주석을 달거나, 채팅을 통한 코드 제안 등
- 좀 더 효율적인 페어프로그래밍이 가능해짐

# Pair Programming



# Unit Test

테스트	기간	특성	오류 메시지
unit_tests (23)	474ms		
<빈 네임스페이스> (15)	474ms		
CleanerTest (4)	< 1ms		
CleanerControllerBusSubscribe...	< 1ms	GoogleTes...	
CleanerHwConnectCheck	< 1ms	GoogleTes...	
TurnOnTurnOffDoesNotThrow	< 1ms	GoogleTes...	
WhenPowerUpTimerCallingCh...	< 1ms	GoogleTes...	
RvcSimulatorTest (3)	474ms		
AutoStepRunsRequestedTicks	474ms	GoogleTes...	
DustTriggersPowerUpAndTime...	< 1ms	GoogleTes...	
StepMovesForwardWhenFrontl...	< 1ms	GoogleTes...	
SensorControllerTest (8)	< 1ms		
TestChecknPowerUp_WhenNo...	< 1ms	GoogleTes...	
TestFrontObstacleDetectedTrue	< 1ms	GoogleTes...	
TestGetLeftStateFalse	< 1ms	GoogleTes...	
TestGetLeftStateTrue	< 1ms	GoogleTes...	
TestGetRightStateFalse	< 1ms	GoogleTes...	
TestGetRightStateTrue	< 1ms	GoogleTes...	
TestTurnOff	< 1ms	GoogleTes...	
TestTurnOn	< 1ms	GoogleTes...	
MotorControllerTest (8)	< 1ms		
MotorControllerAvoidTest (6)	< 1ms		
BothKeep	< 1ms	GoogleTes...	
FrontObstacle	< 1ms	GoogleTes...	
LeftAndRightObstacle	< 1ms	GoogleTes...	
LeftObstacle	< 1ms	GoogleTes...	
RightObstacle	< 1ms	GoogleTes...	
SeqLeftOpen	< 1ms	GoogleTes...	

- Unit Test 총 23개
  - 클리너
  - 센서
  - 모터
- 하나의 Unit Test에서도 여러 값을 비교

```
TEST_F(MotorControllerAvoidTest, RightObstacle) {  
    // 앞, 오른쪽이 막혀있는 상황  
    sensorProvider.left = false;  
    sensorProvider.right = true;  
  
    motorController->AvoidObstacle(sensorProvider);  
  
    ASSERT_TRUE(motor.point.isEqual(Point(-1, 0)));  
    ASSERT_FALSE(motor.point.isEqual(Point(0, 1)));  
    ASSERT_FALSE(motor.point.isEqual(Point(1, 0)));  
    ASSERT_FALSE(motor.point.isEqual(Point(0, -1)));  
  
    ASSERT_EQ(motor.stopCount, 1);  
    ASSERT_EQ(motor.forwardCount, 1);  
    ASSERT_EQ(motor.backwardCount, 0);  
    ASSERT_EQ(motor.leftCount, 1);  
    ASSERT_EQ(motor.rightCount, 0);  
};
```

# Unit Test

파일	테스트 케이스명	검증 내용(요약)
test_cleaner_controller.cpp	TurnOnTurnOffDoesNotThrow	turnOn/turnOff 시 Cleaner HW 상태(isturnOn)가 정상 반영되는지
test_cleaner_controller.cpp	CleanerControllerBusSubscribeCheck	EventBus 이벤트 (StartCleaning, DetectedDust, AvoidObstacle, TurnOff)에 CleanerController가 올바르게 반응하는지
test_cleaner_controller.cpp	WhenPowerUpTimerCallingCheck	CCpowerUp() 후 타이머 5 tick에서 powerRestore가 호출되어 ispowerUp=false로 복귀되는지
test_cleaner_controller.cpp	CleanerHwConnectCheck	CleanerController의 HW 연결 동작(turnOn/off, powerUp/restore)이 직접 호출 시 정상인지

# Unit Test

파일	테스트 케이스명	검증 내용(요약)
test_motor_controller.cpp	TurnOn	전원 ON에서 회피/이동 로직 호출 시 전진·회전 카운트 및 최종 위치가 기대값인지
test_motor_controller.cpp	TurnOff	전원 OFF에서 회피 호출해도 모터 동작(이동/회전)이 발생하지 않는지
test_motor_controller.cpp	FrontObstacle	전방 장애물 조건에서 우회 동작(우회전+전진)이 맞는지
test_motor_controller.cpp	RightObstacle	전방+우측 장애물에서 좌회전 경로 선택이 맞는지
test_motor_controller.cpp	LeftObstacle	전방+좌측 장애물에서 우회전 경로 선택이 맞는지

# Unit Test

파일	테스트 케이스명	검증 내용(요약)
test_motor_controller.cpp	LeftAndRightObstacle	양측 차단 지속 후 우측 개방 시 후진 → 탈출 회전 흐름이 정상인지
test_motor_controller.cpp	BothKeep	좌우 모두 계속 막힌 경우 후진이 지속되는지(장기 반복)
test_motor_controller.cpp	SeqLeftOpen	좌측이 나중에 열리는 시퀀스에서 좌회전 탈출이 되는지
test_sensor_controller.cpp	TestTurnOn	turnOn() 후 내부 ON 상태가 true인지
test_sensor_controller.cpp	TestTurnOff	turnOff() 후 내부 ON 상태가 false인지
test_sensor_controller.cpp	TestFrontObstacleDetectedTrue	FrontObstacleDetected()가 AvoidObstacle 이벤트를 발행하는지

# Unit Test

파일	테스트 케이스명	검증 내용(요약)
test_sensor_controller.cpp	TestChecknPowerUp_WhenNoDust	먼지 미감지 시 DetectedDust 이벤트가 발행되지 않는지
test_sensor_controller.cpp	TestGetLeftStateTrue	좌측 센서 true 반환 확인
test_sensor_controller.cpp	TestGetLeftStateFalse	좌측 센서 false 반환 확인
test_sensor_controller.cpp	TestGetRightStateFalse	우측 센서 false 반환 확인
test_sensor_controller.cpp	TestGetRightStateTrue	우측 센서 true 반환 확인
test_rvc_simulator.cpp	StepMovesForwardWhenFrontIsEmpty	전원 ON 후 1틱 전진 및 Cleaner ON 여부 확인
test_rvc_simulator.cpp	DustTriggersPowerUpAndTimerRestoresPower	먼지 감지 시 PowerUp 진입 + 5 step 후 PowerUp 해제 확인

# Unit Test

파일	테스트 케이스명	검증 내용(요약)
test_rvc_simulator.cpp	TurnOffStopsCleaner	turnOff() 후 전원 OFF 및 Cleaner OFF 확인
test_rvc_simulator.cpp	AutoStepRunsRequested Ticks	autoStep(3) 수행 후 위치가 3틱만큼 이동했는지

# Code Review

Re: [davidpro08/2026\_1\_OOAD] feat: in ... (PR #12) 받은편지함 x

**Kim hokyung** <notifications@github.com> 수신 거부  
davidpro08/2026\_1\_OOAD, 나, Comment에게 ▾

@reload02 approved this pull request.

—  
Reply to this email directly, [view it on GitHub](#), or [unsubscribe](#).  
You are receiving this because you commented.

**Noname0313** <notifications@github.com>  
Push, davidpro08/2026\_1\_OOAD, 나에게 ▾

@nonameS0313 pushed 1 commit.

- [7ac76b8](#) testCase error Detect

—  
[View it on GitHub](#) or [unsubscribe](#).

- Github Ruleset 이용
  - 3인 기준 2인 Approve해야 Merge 가능
  - comment가 있으면 Resolve해야 Merge 가능
- 리뷰 시 이메일 발송하여 확인
- 리뷰 상으로 해결하기 힘든 부분이 있으면 디코 화면 공유나 만나서 해결

nonameS0313 commented 5 hours ago

그럼 타이머도 step 기반?

이게 조금 애매하다고 생각되긴 하는데, 교수님이 계속 OOA, OOD, OOI가 일관되게 해야한다고 해서. 그 관점으로만 생각해! Timer는 그대로 냅두고 ClassDiagram이랑 Domain Model 상에서 Digital Clock과의 연결을 떼버려도 될 거 같은데 어때? Timer를 simulator Step에 연결시켜서 동작하도록 할려면 system Operation이 바뀌는 거라 좀 많이 수정해야할 거 같아. 그냥 Digital Clock과 연결을 떼버리고, 'Timer'라는거는 RVC의 Tick(Digital Clock, 시간 시스템)과 무관하게 자신만의 시간 '다' 라는 느낌으로 가는 거 어때? 별론가?

Timer도 step 기반에 맞추려면 UseCase로 'Tick 동기화' 라는 걸 만들어서 step 마다 Timer의 Tick을 step마다 줄여주는 그런걸 만들고 그걸 바탕으로 코드 짜 거 같아. 생각해보니 그렇게 막 복잡하지는 않을 거 같긴하네. 오히려 코드도 깔끔해지고 구조를 닷? 다이어그램만 요러쿵 저러쿵 수정하면

**Require a pull request before merging**  
Require all commits be made to a non-target branch and submitted via a pull request before they can be merged.

[Hide additional settings](#) ^

**Required approvals**  
2 ▾  
The number of approving reviews that are required before a pull request can be merged.

**Dismiss stale pull request approvals when new commits are pushed**  
New, reviewable commits pushed will dismiss previous pull request review approvals.

시뮬레이터가 우리가 만든 Tick을 안쓰고 직접적으로 SensorController에 접근해서 쓰고 있는데 괜찮나?

- 지금 우리가 OOA, OOD한거를 봤는데, 조금 SystemOperation이랑 OOD랑 안맞는 부분이 있네;; 근데 내가 생각하기로 는 Tick 부분을 그냥 액터로 simulator 영역이라고 바꾸면 simulator도 바꿀 거 없고 그냥 Tick 없애고 Sensor에 if문만 추가하면 될 거 같거든? 어때?

그니까 step 한 번을 Tick이라고 생각하는 거지? 이렇다면 난 괜찮다고 봐

오오 맞아맞아



1

그럼 타이머도 step 기반?

이게 조금 애매하다고 생각되긴 하는데, 교수님이 계속 OOA, OOD, OOI가 일관되게 해야한다고 해서. 그 관점으로만 생각해보면 그냥 Timer는 그대로 냅두고 ClassDiagram이랑 Domain Model 상에서 Digital Clock과의 연결을 떼버려도 될 거 같은데 어때? Timer를 simulator Step에 연결시켜서 동작하도록 할려면 system Opertation이 바뀌는 거라 좀 많이 수정해야할 거 같아. 그냥 Digital Clock과 연결을 떼버리고, 'Timer라는거는 RVC의 Tick(Digital Clock, 시간 시스템)과 무관하게 자신 만의 시간 측정을 한다' 라는 느낌으로 가는 거 어때? 별론가?

Timer도 step 기반에 맞추려면 UseCase로 'Tick 동기화' 라는 걸 만들어서 step 마다 Timer의 Tick을 step마다 줄여주는 그런걸 만들고 그걸 바탕으로 코드를 짜면 될 거 같아. 생각해보니 그렇게 막 복잡하지는 않을 거 같긴하네. 오히려 코드도 깔끔해지고 ㄱ  
ㅈ을 듯? 다이어그램만 요러쿵 저러쿵 수정하면

ㅇㅋㅇㅋ

```
45 - bus.subscribeAvoidObstacle([&isAvoidObstacleCalled](SensorController* sender) {
45 + bus.subscribeAvoidObstacle([&isAvoidObstacleCalled](SensorProvider* sender) {
```



davidpro08 2 hours ago

Owner ...

오 잘 봤네



Reply...

Unresolve conversation

davidpro08 marked this conversation as resolved.

2026\_1\_OOAD/Simulator/RvcSimulator.cpp

Hide resolved

```
88 89
89 90     motor.clearBlocked();
90 - if (frontSensor.detect()) {
91 + if (frontSensor.detect() && !motorController.avoiding) {
```



davidpro08 1 hour ago

Owner ...

후진 상태 추가 Simulator를 보여주려면 좋은 방법인듯



Reply...

---

# SYSTEM TEST

---

# Simulator

```
C:\Users\문재현\source\Wrej x +
==== RVC Simulator ====
1. 기본 모드
2. 테스트 모드
3. 종료
선택>
```

```
RVC CLI 격자 시뮬레이터

명령 목록
on          전원 켜기
off         전원 끄기
step       1틱 진행
auto N     N틱 자동 진행
front      전방 장애물 이벤트 강제 발생
dust x y   지정 칸에 먼지 추가
wall x y   지정 칸 벽 토글
status     현재 맵과 상태 출력
reset      기본 맵으로 초기화
randmap [시드] 무작위 맵(시드 생략 시 매번 다름)
help       도움말 출력
quit       종료

#####
#         #
#   *   # #
# #####*# #
# *     #
#         #
#     ## #
# ###   * #
# # # *  #
# # #   #
# ^     #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (1, 1) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 1, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

명령>
```

- 1번 선택 시 명령어 콘솔
- 시작 메뉴
  - 1: 기본 모드
  - 2: 테스트 모드
  - 3 / q / quit: 종료
- 기본 모드 입력
  - on, off, step, status, help, reset, front
  - auto N (예: auto 10)
  - dust x y (예: dust 3 4)
  - wall x y (예: wall 5 6)
  - randmap 또는 randmap seed (예: randmap 123)
  - quit / exit (기본 모드 나가기)

# Simulator

```
==== RVC Simulator ====
1. 기본 모드
2. 테스트 모드
3. 종료
선택> 2

System Test Mode (1~30, A: 전체, Q: 종료)
test> 1
---- Case 1 Start ----
[Positive] TurnOff 일때 움직이지 않는가

#####
#   #
#   #
#   #
# ^ #
#   #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#   #
#   #
#   #
# ^ #
#   #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 1 End ----
[Case 1][Positive] PASS - TurnOff 일때 움직이지 않는가
Total: 1, Pass: 1, Fail: 0
test>
```

- 2번 선택 시 테스트 콘솔
- 테스트 모드 입력
  - A / a / all: 전체 테스트
  - 1~30: 해당 번호 테스트
  - Q / q / quit: 테스트 모드 나가기

# System Test

ID	유형	테스트명	핵심 검증
1	Positive	TurnOff 일때 움직이지 않는가	전원 OFF에서 step() 시 위치 변화 없음
2	Positive	앞에 장애물이 없을때 앞으로 가 가	1틱 전진 위치 일치
3	Positive	여러 Tick걸쳐 앞으로 전진하는가	2틱 누적 전진 정상
4	Positive	장애물피하기 중 Cleaning 중단	Avoid 중 Cleaner OFF
5	Positive	전방 장애물, 양측 오픈 시 우회전	우회전 방향 전환
6	Positive	전방+우측 장애물 시 좌회전	좌회전 방향 전환
7	Positive	후진 중 우측 길 발견 시 회전	우측 탈출 회전
8	Positive	후진 중 좌측 길 발견 시 회전	좌측 탈출 회전

# System Test

ID	유형	테스트명	핵심 검증
9	Positive	Avoid 후 전진/청소 재개	회피 후 이동 재개 + Cleaner ON
10	Positive	꼬부랑 길에서 에러 없이 동작	장시간 진행 중 전원 유지
11	Positive	Dust 발견 시 PowerUp	먼지 감지 후 PowerUp 진입
12	Positive	PowerUp 중 Dust 재발견 처리	PowerUp 유지 동작

# System Test

ID	유형	테스트명	핵심 검증
13	Negative	우측 센서 고장	우측 미검출 고장 시 충돌/차단 재현
14	Negative	Dust 음수 좌표 입력	addDust(-1,-1) 거부
15	Negative	시작 즉시 3면 차단	차단 상황에서도 시스템 유지
16	Negative	후진 무한 경로에서 제한 동작	연속 Avoid 카운트 제한 확인
17	Negative	모터 고장 시 움직이지 않음	모터 고장 시 비정상 이동 방지
18	Negative	동작 중 전원 끄기	OFF 후 추가 step()에도 정지
19	Negative	Avoid 후 PowerUp 잔여 유지	Avoid 상태에서 PowerUp 잔여 확인
20	Negative	후진 중 좌우 센서 고장	좌우 센서 고장 시 폴백 동작

# System Test

ID	유형	테스트명	핵심 검증
21	Negative	후진 중 우측 센서 노이즈	노이즈 상황에서 진행/복구
22	Negative	Dust 센서 항상 high	always-high에서 PowerUp 반응
23	Negative	Dust 센서 항상 0	always-zero에서 PowerUp 미진입
24	Negative	음수 좌표 입력 거부	addDust/toggleWall 음수 거부
25	Negative	맵 경계 밖 좌표 입력 거부	경계 밖 좌표 거부
26	Negative	시작 위치가 벽이면 로드 실패	벽 위 시작점 시나리오 로드 실패
27	Negative	시작 위치가 맵 밖이면 로드 실패	맵 밖 시작점 로드 실패
28	Negative	행 길이 불일치 맵 입력 거부	비정형 맵 로드 거부

# System Test

ID	유형	테스트명	핵심 검증
29	Negative	벽/먼지 동일 좌표 충돌 처리	벽 우선, 벽 위 먼지 추가 거부
30	Negative	먼지 위 시작 위치 시나리오 처리	먼지 위 시작 후 기본 동작 유지

# Simulator

```
==== RVC Simulator ====
1. 기본 모드
2. 테스트 모드
3. 종료
선택> 2

System Test Mode (1~30, A: 전체, Q: 종료)
test> 1
---- Case 1 Start ----
[Positive] TurnOff 일때 움직이지 않는가

#####
#   #
#   #
#   #
# ^ #
#   #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#   #
#   #
#   #
# ^ #
#   #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 1 End ----
[Case 1][Positive] PASS - TurnOff 일때 움직이지 않는가
Total: 1, Pass: 1, Fail: 0
test>
```

- 2번 선택 시 테스트 콘솔
- 테스트 모드 입력
  - A/a/all: 전체 테스트
  - 1~30: 해당 번호 테스트
  - Q/q/quit: 테스트 모드 나가기

# System Test

```
System Test Mode (1~30, A: 전체, Q: 종료)
test> 1
---- Case 1 Start ----
[Positive] TurnOff 일때 움직이지 않는가

#####
#      #
#      #
#      #
# ^    #
#      #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (2, 2) | 방향: 위 (0, 1)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 0, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#      #
#      #
#      #
# ^    #
#      #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (2, 2) | 방향: 위 (0, 1)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 0, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 1 End ----
[Case 1][Positive] PASS - TurnOff 일때 움직이지 않는가
Total: 1, Pass: 1, Fail: 0
test>
```

```
---- Case 2 Start ----
[Positive] 앞에 장애물이 없을때 앞으로 가는가

#####
#      #
#      #
#      #
# ^    #
#      #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (2, 2) | 방향: 위 (0, 1)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 0, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#      #
#      #
#      #
# ^    #
#      #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (2, 3) | 방향: 위 (0, 1)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 0, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 2 End ----
[Case 2][Positive] PASS - 앞에 장애물이 없을때 앞으로 가는가
Total: 1, Pass: 1, Fail: 0
test>
```

# System Test

```
#####
#      #
#      #
#      #
# ^    #
#      #
#      #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (2, 3) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#      #
#      #
# ^    #
#      #
#      #
#      #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (2, 4) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 3 End ----
[Case 3][Positive] PASS - 여러 Tick걸쳐 앞으로 전진하는가
Total: 1, Pass: 1, Fail: 0
test>
```

```
test> 4
---- Case 4 Start ----
[Positive] 장애물피하기 중 Cleaning 중단

#####
#      #
#      #
# #    #
# ^    #
#      #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 1, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#      #
#      #
# #    #
# >   #
#      #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (3, 2) | 방향 : 오른쪽 (1, 0)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 4 End ----
[Case 4][Positive] FAIL - Avoid 중 Cleaner ON
Total: 1, Pass: 0, Fail: 1
test>
```

# System Test

```
test> 5
---- Case 5 Start ----
[Positive] 전방 장애물, 양측 오픈 시 우회전

#####
#      #
#      #
# # #
# ^ #
#      #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (3, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 1, 좌측벽 : 0, 우측벽 : 0, 현재면지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#      #
#      #
# # #
# > #
#      #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (4, 2) | 방향 : 오른쪽 (1, 0)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재면지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 5 End ----
[Case 5][Positive] PASS - 전방 장애물, 양측 오픈 시 우회전
Total: 1, Pass: 1, Fail: 0
test> |
```

```
---- Case 6 Start ----
[Positive] 전방+우측 장애물 시 좌회전

#####
#      #
#      #
# # #
# ^# #
#      #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (3, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 1, 좌측벽 : 0, 우측벽 : 1, 현재면지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#      #
#      #
# # #
# < # #
#      #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (2, 2) | 방향 : 왼쪽 (-1, 0)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재면지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 6 End ----
[Case 6][Positive] PASS - 전방+우측 장애물 시 좌회전
Total: 1, Pass: 1, Fail: 0
test>
```

# System Test

```
#####
#   #
#   #
#  # #
# #^# #
#   #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (3, 2) | 방향: 위 (0, 1)
센서 - 전방벽: 1, 좌측벽: 1, 우측벽: 1, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#   #
#   #
#  # #
# # # #
# > #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (3, 1) | 방향: 오른쪽 (1, 0)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 1, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#   #
#   #
#  # #
# # # #
# > #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (4, 1) | 방향: 오른쪽 (1, 0)
센서 - 전방벽: 0, 좌측벽: 1, 우측벽: 1, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 7 End ----
[Case 7][Positive] PASS - 후진 중 우측 길 발견 시 회전
Total: 1, Pass: 1, Fail: 0
test> |
```

```
#####
#   #
#   #
#  # #
# #^# #
#   #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (3, 2) | 방향: 위 (0, 1)
센서 - 전방벽: 1, 좌측벽: 1, 우측벽: 1, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#   #
#   #
#  # #
# # # #
# <# #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (3, 1) | 방향: 왼쪽 (-1, 0)
센서 - 전방벽: 0, 좌측벽: 1, 우측벽: 0, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#   #
#   #
#  # #
# # # #
# <# #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (2, 1) | 방향: 왼쪽 (-1, 0)
센서 - 전방벽: 0, 좌측벽: 1, 우측벽: 1, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 8 End ----
[Case 8][Positive] PASS - 후진 중 좌측 길 발견 시 회전
Total: 1, Pass: 1, Fail: 0
test> |
```

# System Test

```
[Positive] Avoid 후 전진/청소 재개

#####
# #
# #
# # #
# ^ #
# #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (3, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 1, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# # #
# > #
# #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (4, 2) | 방향 : 오른쪽 (1, 0)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# # #
# >#
# #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (5, 2) | 방향 : 오른쪽 (1, 0)
센서 - 전방벽 : 1, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 9 End ----
[Case 9][Positive] PASS - Avoid 후 전진/청소 재개
Total: 1, Pass: 1, Fail: 0
test>
```

```
센서 - 전방벽 : 1, 좌측벽 : 1, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# # #
# #### #
# # #
# # #
# < #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (6, 1) | 방향 : 왼쪽 (-1, 0)
센서 - 전방벽 : 0, 좌측벽 : 1, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# # #
# #### #
# # #
# # #
# < #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (5, 1) | 방향 : 왼쪽 (-1, 0)
센서 - 전방벽 : 0, 좌측벽 : 1, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 10 End ----
[Case 10][Positive] PASS - 꼬부랑 길에서 에러 없이 동작
Total: 1, Pass: 1, Fail: 0
test>
```

# System Test

```
test> 11
---- Case 11 Start ----
[Positive] Dust 발견 시 PowerUp

#####
#      #
#      #
# *    #
# ^    #
#      #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#      #
#      #
# ^    #
#      #
#      #
#####
전원 : ON | 청소기 : ON | 파워업 : ON
위치 : (2, 3) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

---- Case 11 End ----
[Case 11][Positive] PASS - Dust 발견 시 PowerUp
Total: 1, Pass: 1, Fail: 0
test>
```

```
#      #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#      #
#      #
# *    #
# ^    #
#      #
#      #
#####
전원 : ON | 청소기 : ON | 파워업 : ON
위치 : (2, 3) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#      #
#      #
# ^    #
# x    #
#      #
#      #
#####
전원 : ON | 청소기 : ON | 파워업 : ON
위치 : (2, 4) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 12 End ----
[Case 12][Positive] PASS - PowerUp 중 Dust 재발견 처리
Total: 1, Pass: 1, Fail: 0
test>
```

# System Test

```
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 12 End ----
[Case 12][Positive] PASS - PowerUp 중 Dust 재발견 처리
Total: 1, Pass: 1, Fail: 0
test> 13
---- Case 13 Start ----
[Negative] 우측 센서 고장

#####
# #
# #
# # #
# ^# #
# #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (3, 2) | 방향: 위 (0, 1)
센서 - 전방벽: 1, 좌측벽: 0, 우측벽: 1, 현재먼지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# # #
# ># #
# #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (3, 2) | 방향: 오른쪽 (1, 0)
센서 - 전방벽: 1, 좌측벽: 1, 우측벽: 0, 현재먼지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 13 End ----
[Case 13][Negative] PASS - 우측 센서 고장
Total: 1, Pass: 1, Fail: 0
test>
```

```
#####
# #
# #
# # #
# ># #
# #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (3, 2) | 방향: 오른쪽 (1, 0)
센서 - 전방벽: 1, 좌측벽: 1, 우측벽: 0, 현재먼지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 13 End ----
[Case 13][Negative] PASS - 우측 센서 고장
Total: 1, Pass: 1, Fail: 0
test> 14
---- Case 14 Start ----
[Negative] Dust 음수 좌표 입력

#####
# #
# #
# #
#^ #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (1, 1) | 방향: 위 (0, 1)
센서 - 전방벽: 0, 좌측벽: 1, 우측벽: 0, 현재먼지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 14 End ----
[Case 14][Negative] PASS - Dust 음수 좌표 입력
Total: 1, Pass: 1, Fail: 0
test>
```

# System Test

```
----- Case 15 Start -----  
[Negative] 시작 즉시 3면 차단  
  
#####  
# #  
# #  
# # #  
# #^# #  
# #  
#####  
전원: OFF | 청소기: OFF | 파워업: OFF  
위치: (3, 2) | 방향: 위 (0, 1)  
센서 - 전방벽: 1, 좌측벽: 1, 우측벽: 1, 현재면지: 0  
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC  
  
#####  
# #  
# #  
# # #  
# # # #  
# > #  
#####  
전원: ON | 청소기: ON | 파워업: OFF  
위치: (3, 1) | 방향: 오른쪽 (1, 0)  
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 1, 현재면지: 0  
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC  
----- Case 15 End -----  
[Case 15][Negative] PASS - 시작 즉시 3면 차단  
Total: 1, Pass: 1, Fail: 0  
test>
```

```
#####  
#< #  
# #  
# #  
# # #  
# # # #  
# #  
#####  
전원: ON | 청소기: ON | 파워업: OFF  
위치: (1, 8) | 방향: 왼쪽 (-1, 0)  
센서 - 전방벽: 1, 좌측벽: 0, 우측벽: 1, 현재면지: 0  
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC  
  
#####  
# #  
#v #  
# #  
# #  
# #  
# # #  
# # # #  
# #  
#####  
전원: ON | 청소기: ON | 파워업: OFF  
위치: (1, 7) | 방향: 아래 (0, -1)  
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 1, 현재면지: 0  
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC  
----- Case 16 End -----  
[Case 16][Negative] FAIL - 후진 상태/제한 카운트가 명  
Total: 1, Pass: 0, Fail: 1  
test>
```

# System Test

```
test> 17
---- Case 17 Start ----
[Negative] 모터 고장 시 움직이지 않음

#####
# #
# #
# #
# ^ #
# #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재면지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# #
# ^ #
# #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재면지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 17 End ----
[Case 17][Negative] PASS - 모터 고장 시 움직이지 않음
Total: 1, Pass: 1, Fail: 0
test>
```

```
#####
# #
# #
# #
# ^ #
# #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재면지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# ^ #
# #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (2, 3) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재면지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# ^ #
# #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 3) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재면지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 18 End ----
[Case 18][Negative] PASS - 동작 중 전원 끄기
Total: 1, Pass: 1, Fail: 0
test>
```

# System Test

```
#####
# #
# #
# * #
# ^ #
# #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (2, 2) | 방향: 위 (0, 1)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 0, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# ^ #
# #
# #
#####
전원: ON | 청소기: ON | 파워업: ON
위치: (2, 3) | 방향: 위 (0, 1)
센서 - 전방벽: 1, 좌측벽: 0, 우측벽: 0, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# x> #
# #
# #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (3, 3) | 방향: 오른쪽 (1, 0)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 0, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 19 End ----
[Case 19][Negative] FAIL - Avoid 상태에서 PowerUp 잔
Total: 1, Pass: 0, Fail: 1
test>
```

```
[Negative] 후진 중 좌우 센서 고장

#####
# #
# #
# #
# #^#
# #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (3, 2) | 방향: 위 (0, 1)
센서 - 전방벽: 1, 좌측벽: 1, 우측벽: 1, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# #
# # #
# # #
# > #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (3, 1) | 방향: 오른쪽 (1, 0)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 1, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# #
# #
# #
# # #
# # #
# > #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (4, 1) | 방향: 오른쪽 (1, 0)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 0, 현재면지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 20 End ----
[Case 20][Negative] PASS - 후진 중 좌우 센서 고장
Total: 1, Pass: 1, Fail: 0
test>
```

# System Test

```
#####
#   #
#   #
#  ^#
# # #
#   #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (5, 3) | 방향: 위 (0, 1)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 1, 현재먼지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#   #
#  ^#
# # #
# # #
#   #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (5, 4) | 방향: 위 (0, 1)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 1, 현재먼지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#  ^#
#   #
# # #
# # #
#   #
#####
전원: ON | 청소기: ON | 파워업: OFF
위치: (5, 5) | 방향: 위 (0, 1)
센서 - 전방벽: 1, 좌측벽: 0, 우측벽: 1, 현재먼지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 21 End ----
[Case 21][Negative] PASS - 후진 중 우측 센서 노이즈
Total: 1, Pass: 1, Fail: 0
test>
```

```
#####
# ^ #
#   #
# x #
#   #
#   #
#####
전원: ON | 청소기: ON | 파워업: ON
위치: (2, 5) | 방향: 위 (0, 1)
센서 - 전방벽: 1, 좌측벽: 0, 우측벽: 0, 현재먼지: 1
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# > #
#   #
# x #
#   #
#   #
#####
전원: ON | 청소기: ON | 파워업: ON
위치: (3, 5) | 방향: 오른쪽 (1, 0)
센서 - 전방벽: 0, 좌측벽: 1, 우측벽: 0, 현재먼지: 1
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# > #
#   #
# x #
#   #
#   #
#####
전원: ON | 청소기: ON | 파워업: ON
위치: (4, 5) | 방향: 오른쪽 (1, 0)
센서 - 전방벽: 0, 좌측벽: 1, 우측벽: 0, 현재먼지: 1
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 22 End ----
[Case 22][Negative] PASS - Dust 센서 항상 high
Total: 1, Pass: 1, Fail: 0
test>
```

# System Test

```
---- Case 22 End ----
[Case 22][Negative] PASS - Dust 센서 항상 high
Total: 1, Pass: 1, Fail: 0
test> 23
---- Case 23 Start ----
[Negative] Dust 센서 항상 0

#####
#   #
#   #
# *  #
# ^  #
#   #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
#   #
#   #
# ^  #
#   #
#   #
#####
전원 : ON | 청소기 : ON | 파워업 : OFF
위치 : (2, 3) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 23 End ----
[Case 23][Negative] PASS - Dust 센서 항상 0
Total: 1, Pass: 1, Fail: 0
test>
```

```
---- Case 24 Start ----
[Negative] 맵 경계 밖 좌표

#####
#   #
#   #
#   #
# ^  #
#   #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (2, 2) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 24 End ----
[Case 24][Negative] PASS - 맵 경계 밖 좌표
Total: 1, Pass: 1, Fail: 0
test> |
```

# System Test

```
---- Case 25 Start ----  
[Negative] 맵 경계 밖 좌표 입력 거부  
  
#####  
# #  
# #  
#^ #  
#####  
전원 : OFF | 청소기 : OFF | 파워업 : OFF  
위치 : (1, 1) | 방향 : 위 (0, 1)  
센서 - 전방벽 : 0, 좌측벽 : 1, 우측벽 : 0, 현재먼지 : 0  
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC  
---- Case 25 End ----  
[Case 25][Negative] PASS - 맵 경계 밖 좌표 입력 거부  
Total: 1, Pass: 1, Fail: 0  
test>
```

```
---- Case 26 Start ----  
[Negative] 시작 위치가 벽이면 로드 실패  
  
#####  
# #  
# #  
#^ #  
#####  
전원 : OFF | 청소기 : OFF | 파워업 : OFF  
위치 : (1, 1) | 방향 : 위 (0, 1)  
센서 - 전방벽 : 0, 좌측벽 : 1, 우측벽 : 0, 현재먼지 : 0  
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC  
---- Case 26 End ----  
[Case 26][Negative] PASS - 시작 위치가 벽이면 로드 실패  
Total: 1, Pass: 1, Fail: 0  
test>
```

# System Test

```
---- Case 27 Start ----
[Negative] 시작 위치가 맵 밖이면 로드 실패

#####
#   #
#   #
#   #
# ^  #
#   #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (2, 2) | 방향: 위 (0, 1)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 0, 현재먼지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 27 End ----
[Case 27][Negative] PASS - 시작 위치가 맵 밖이면 로드 실패
Total: 1, Pass: 1, Fail: 0
test>
```

```
---- Case 28 Start ----
[Negative] 행 길이가 불일치 맵 입력 거부

#####
#   #
#   #
#   #
# ^  #
#   #
#####
전원: OFF | 청소기: OFF | 파워업: OFF
위치: (2, 2) | 방향: 위 (0, 1)
센서 - 전방벽: 0, 좌측벽: 0, 우측벽: 0, 현재먼지: 0
범례: # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 28 End ----
[Case 28][Negative] PASS - 행 길이가 불일치 맵 입력 거부
Total: 1, Pass: 1, Fail: 0
test>
```

# System Test

```
---- Case 29 End ----
[Case 29][Negative] PASS - 벽/먼지 동일 좌표 충돌 처리
Total: 1, Pass: 1, Fail: 0
test> 30
---- Case 30 Start ----
[Negative] 먼지 위 시작 위치 시나리오 처리

#####
#      #
#      #
#    #  #
#      #
#      #
#^     #
#####
전원 : OFF | 청소기 : OFF | 파워업 : OFF
위치 : (1, 1) | 방향 : 위 (0, 1)
센서 - 전방벽 : 0, 좌측벽 : 1, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC

#####
# ^ #
# x #
#   #
#####
전원 : ON | 청소기 : ON | 파워업 : ON
위치 : (2, 3) | 방향 : 위 (0, 1)
센서 - 전방벽 : 1, 좌측벽 : 0, 우측벽 : 0, 현재먼지 : 0
범례 : # 벽, . 빈칸, * 먼지, x 청소완료, ^v<> RVC
---- Case 30 End ----
[Case 30][Negative] PASS - 먼지 위 시작 위치 시나리오 처리
Total: 1, Pass: 1, Fail: 0
test>
```

---

# SYSTEM CODE ANALYSIS

---

# Code Analysis

- Github Action을 활용
- 로컬에서 cppcheck, clang-tidy 수행해서 오류 없는지 확인
- Github Action으로 서버에서 cppcheck, clang-tidy 수행해서 오류 확인
  - master branch에 push 막아놔기 때문에 오류가 있는 코드가 강제로 master로 들어가는 걸 방지

```
analyze-and-test:
  uses: actions/checkout@v4

  name: Install Dependencies
  run: |
    sudo apt-get update
    sudo apt-get install -y cmake cppcheck clang-tidy libgtest-dev

  name: Configure CMake
  # CMake 구성 단계에서 Cppcheck와 Clang-Tidy가 활성화됩니다.
  run: |
    cmake -S . -B build \
      -DCMAKE_CXX_CPPCHECK="cppcheck;--enable=all;--inconclusive" \
      -DCMAKE_CXX_CLANG_TIDY="clang-tidy;-header-filter=.*"

  name: Build Project
  # 빌드 과정에서 Cppcheck와 Clang-Tidy 분석이 함께 수행됩니다.
  run: cmake --build build

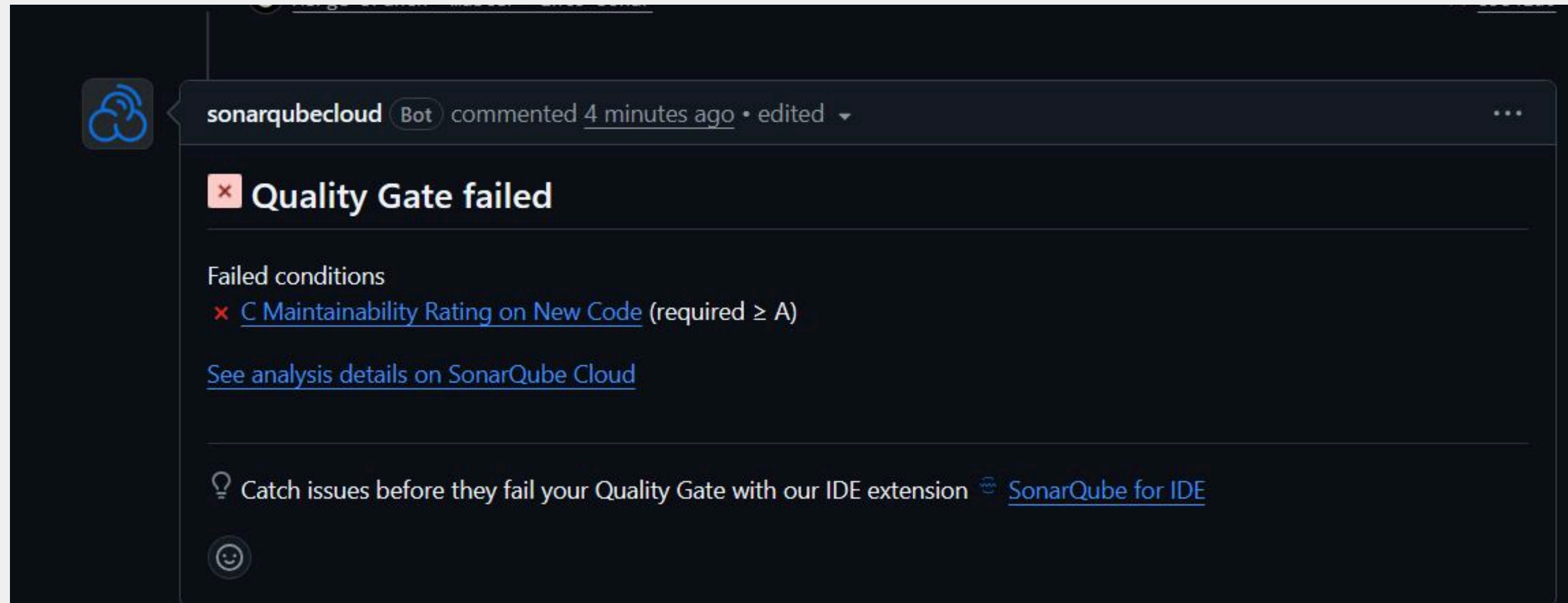
  name: Run Google Test
  # 빌드된 테스트 실행 파일을 실행합니다.
  run: |
    cd build
    ctest --output-on-failure
```

## analyze-and-test

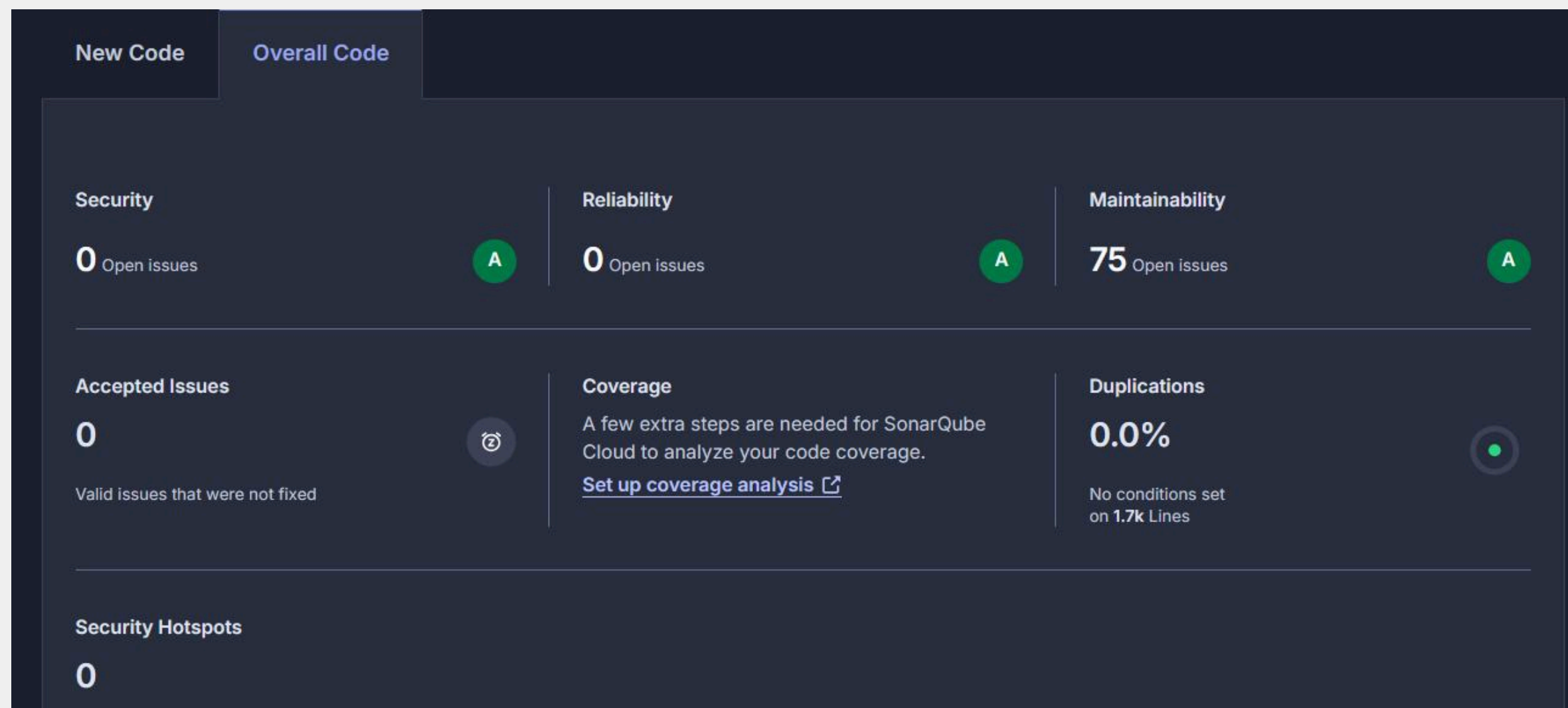
succeeded 2 hours ago in 2m 33s

- > ✓ Set up job
- > ✓ Run actions/checkout@v4
- > ✓ Install Dependencies
- > ✓ Configure CMake
- > ✓ Build Project
- > ✓ Run Google Test
- > ✓ Post Run actions/checkout@v4
- > ✓ Complete job

# Sonarqube



- Sonarqube(cloud) 사용
- Github와 연동해서 Quality가 일정 이상이 아닌 경우 Pull Request 거부
- 코드 개선을 통해 Security, Reliability, Maintainability 를 모두 A 등급을 맞게 해서 통과
- 커버리지는 하지 않음 → 통과하려면 Simulator에 대한 Test Code가 있어야 하는데, 이는 명세 위반이라고 생각
  - 마지막에는 전체 커버리지가 60%정도 나왔음



# Code Smell

2026\_1\_OOAD/Simulator/GridMap.cpp

- Do not use the constructor's initializer list for data member "width". Use the in-class initializer instead. **Intentionality**  
Maintainability **Medium** cppcoreguidelin... performance +  
Open Not assigned L13 · 10min effort · 1 day ago · Code Smell · Major
- Do not use the constructor's initializer list for data member "height". Use the in-class initializer instead. **Intentionality**  
Maintainability **Medium** cppcoreguidelin... performance +  
Open Not assigned L13 · 10min effort · 1 day ago · Code Smell · Major
- Replace with the version of "std::ranges::shuffle" that takes a range. **Consistency**  
Maintainability **Low** bad-practice clumsy ... +  
Open 문제현 Moon Jaehyun L82 · 2min effort · 3 hours ago · Code Smell · Minor
- Replace the redundant type with "auto". **Intentionality**  
Maintainability **Medium** clumsy cppcoreguidelin... ... +  
Open 문제현 Moon Jaehyun L86 · 2min effort · 3 hours ago · Code Smell · Major

- Merge가 안 된 대표적인 이유
- Maintainability 쪽에서 많은 Code Smell이 있었으며, 해결하기 전 까지 Merge가 불가능했음
  - 큰 메소드 → 작은 메소드 여러 개로 나눔
  - 매직 넘버 제거
- 위 노력 등으로 인해 정적 분석에 통과하고 병합 가능

Adaptability | Not focused

**Refactor this code to not nest more than 3 if|for|do|while|switch statements.**

Control flow statements "if", "for", "while", "switch" and "try" should not be nested too deeply [cpp:S134](#)

Software qualities impacted: Maintainability **High**

Open 문제현 Moon Jaehyun Code Smell Critical

Tags: brain-overload

Line affected: L191

Effort: 10 min

Introduced: 3 hours ago

Where is the issue? Why is this an issue? How can I fix it? Activity More info

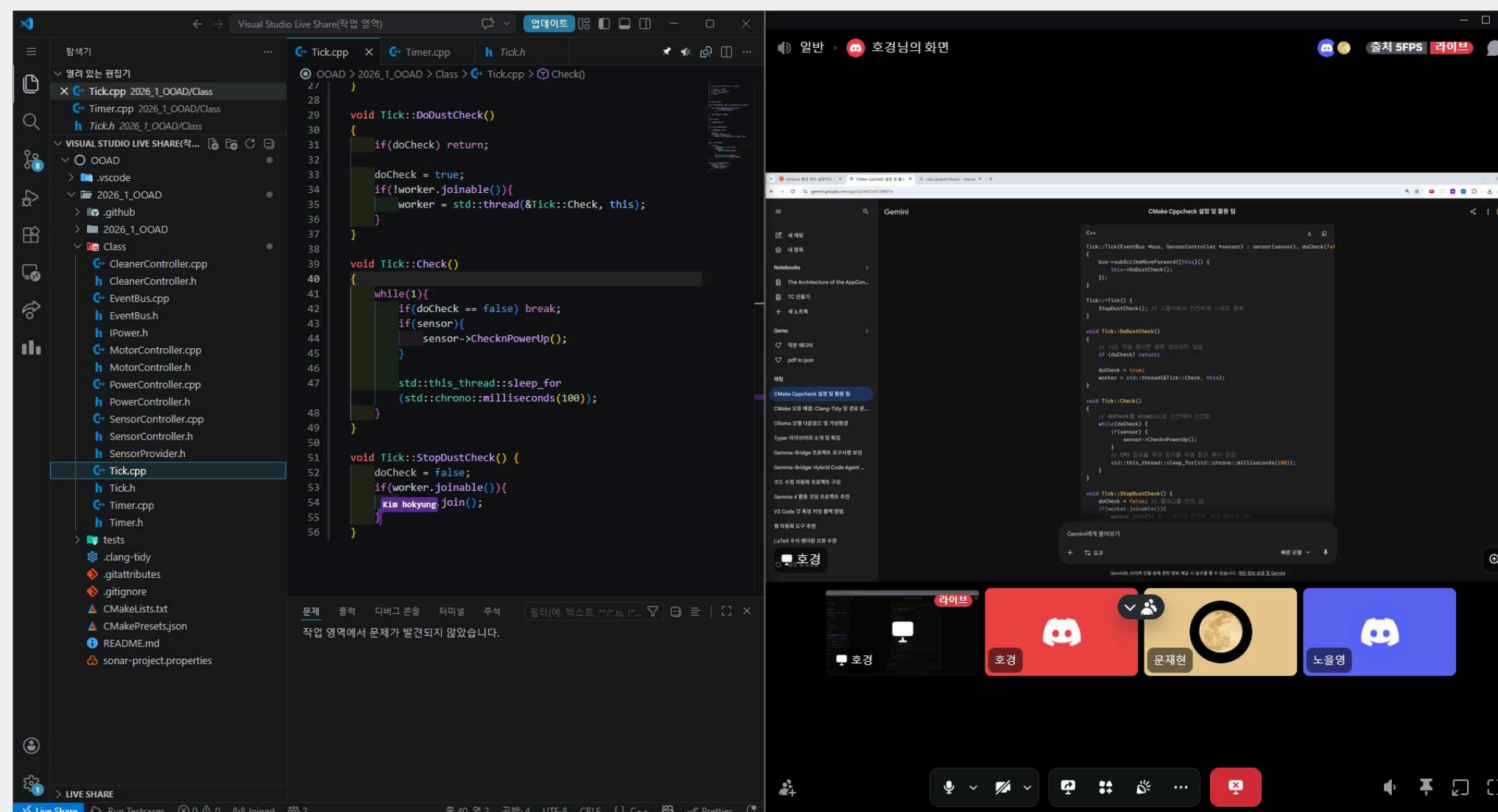
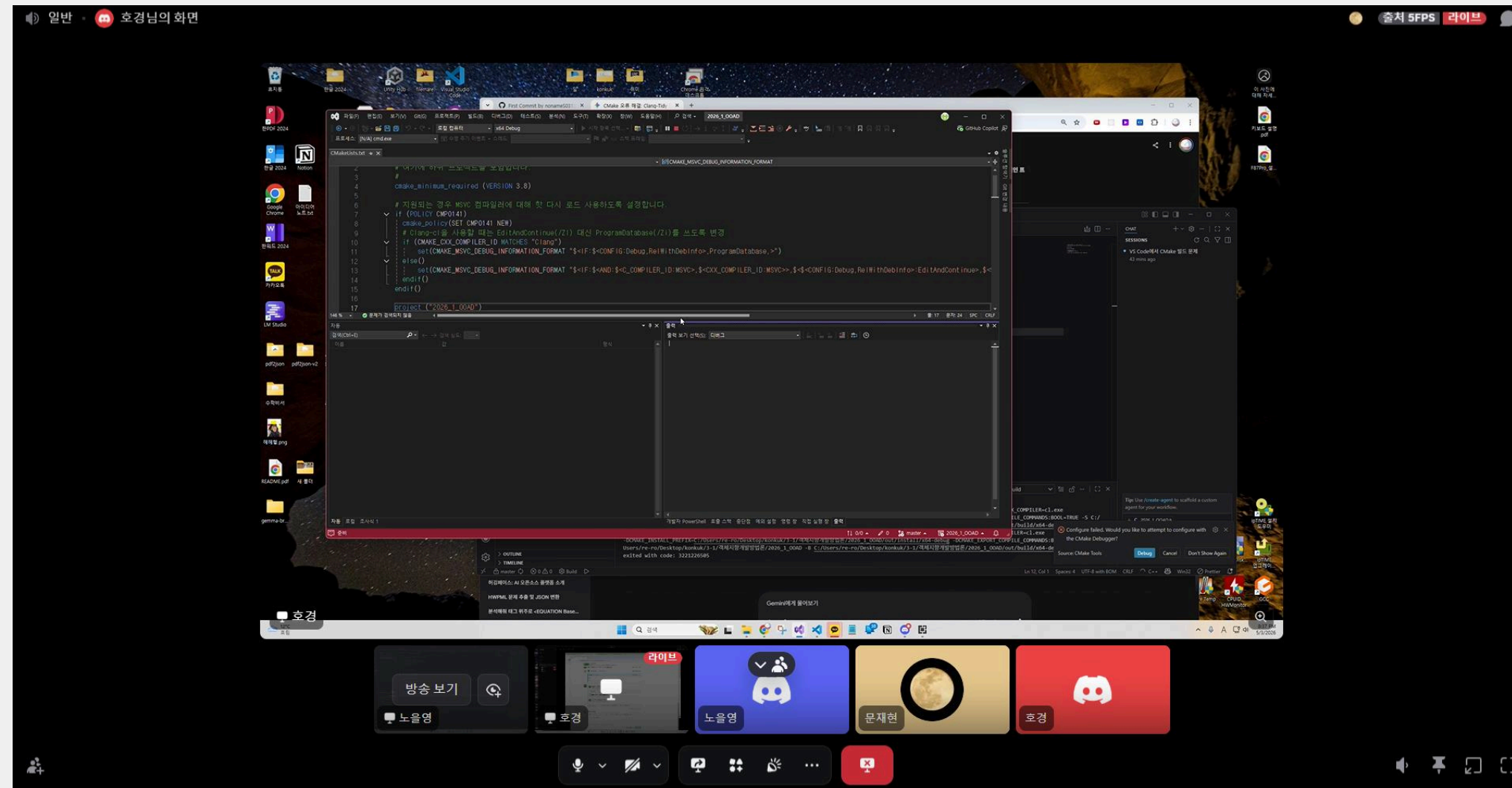
Open in IDE

---

# SUMMARY & TROUBLE SHOOTING

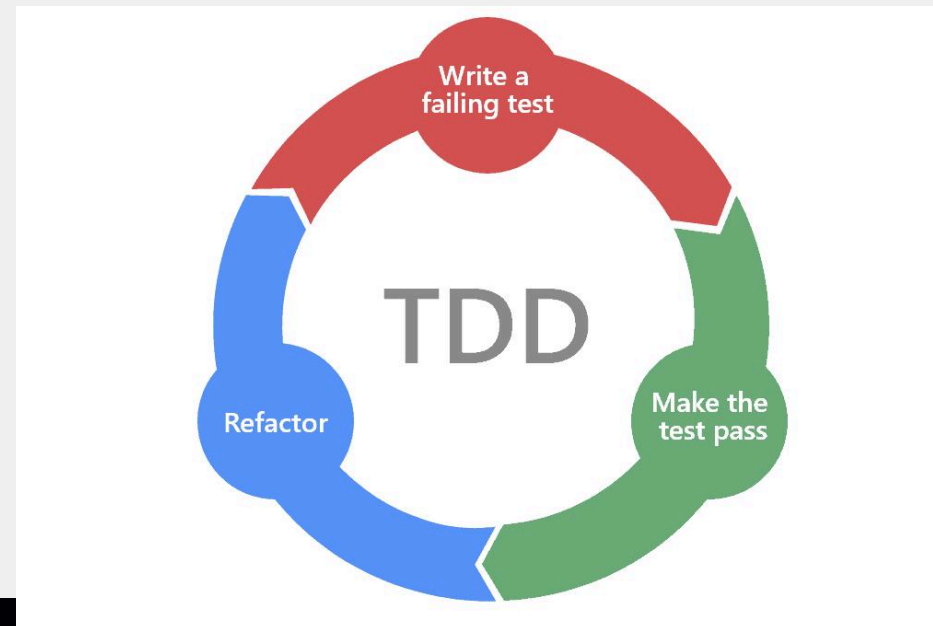
---

# Pair Programming



- 명세를 보지 않고 코드를 짜는 경험
- 말만 듣고 코드를 짜다보니 서로의 생각이 조금씩 다르다는 걸 깨달음
  - 문서화의 중요성
  - 생각을 맞춰가는 과정이 중요
  - 분쟁 발생 시 두 가지 모두 구현하고 더 좋은 방식으로 해결
- 서로가 잘 하는 지점을 이해할 수 있었음
  - 코드를 잘 짜는 사람(단축키 이용, 언어 이해도)
  - 설명을 잘 하는 사람(명세에 대한 이해, 논리적 사고)
  - Git을 잘 다루는 사람(깃 명령어)
- 이후 분업해서 작업을 할 때도 모르는 분야가 있으면 잘 하는 사람에게 질문해서 해결

# Unit Test



```
9/21 Test #9: MotorControllerAvoidTest.BothKeep ..... Passed 0.00 sec
  Start 10: MotorControllerAvoidTest.SeqLeftOpen
10/21 Test #10: MotorControllerAvoidTest.SeqLeftOpen ..... Passed 0.00 sec
  Start 11: MotorControllerAvoidTest.EventAvoid
11/21 Test #11: MotorControllerAvoidTest.EventAvoid .....***Failed 0.00 sec
Running main() from /home/runner/work/2026_1_OOAD/2026_1_OOAD/build/_deps/googletest-src/goog
Note: Google Test filter = MotorControllerAvoidTest.EventAvoid
[=====] Running 1 test from 1 test suite.
[-----] Global test environment set-up.
[-----] 1 test from MotorControllerAvoidTest
[ RUN    ] MotorControllerAvoidTest.EventAvoid
/home/runner/work/2026_1_OOAD/2026_1_OOAD/tests/test_motor_controller.cpp:273: Failure
Expected equality of these values:
  motor.rightCount
  Which is: 0
1
[ FAILED ] MotorControllerAvoidTest.EventAvoid (0 ms)
[-----] 1 test from MotorControllerAvoidTest (0 ms total)

[-----] Global test environment tear-down
[=====] 1 test from 1 test suite ran. (0 ms total)
[ PASSED ] 0 tests.
[ FAILED ] 1 test, listed below:
[ FAILED ] MotorControllerAvoidTest.EventAvoid

1 FAILED TEST

  Start 12: MotorControllerAvoidTest.EventForward
12/21 Test #12: MotorControllerAvoidTest.EventForward ..... Passed 0.00 sec
```

- Test를 실행하며 코드 상의 문제가 많다는 것을 깨달음
  - 구현의 문제 → 구현이 잘못 된 거여서 명세가 안바뀜
    - MCStop() 을 안 했다거나, EventListener 등록 순서가 잘못되었다거나 등등
  - 구조의 문제 → 근본적으로 구조가 잘못된 거여서 어쩔 수 없이 몇 가지는 바뀌어야 했음
    - 후진 상태를 저장해야 할 필요성이 생김 (시뮬레이터와의 연동 + 기다리는 동작이 필요해졌음)
- TDD의 Red-Green-Blue 테스트를 일부 수행
  - Red: 실패하는 테스트 코드 만들기 (구현 부 없음)
  - Green: 최대한 빨리 통과하는 구현 코드 만들기
  - Blue: 코드 스멜 등을 잡아내고 리팩토링하기

# Trouble Shooting 1

Adaptability | Not focused

Refactor this code to not nest more than 3 if|for|do|while|switch statements. [↗](#)

Control flow statements "if", "for", "while", "switch" and "try" should not be nested too deeply [cpp:S134](#)

Software qualities impacted: Maintainability ● High

Open ▼ 문제현 Moon Jaehyun ▼ Code Smell ● Critical

Where is the issue? Why is this an issue? How can I fix it?

```
std::vector<std::string> GridMap::render(Point robot, Point direction) const {
    std::vector<std::string> lines;
    lines.reserve(height);

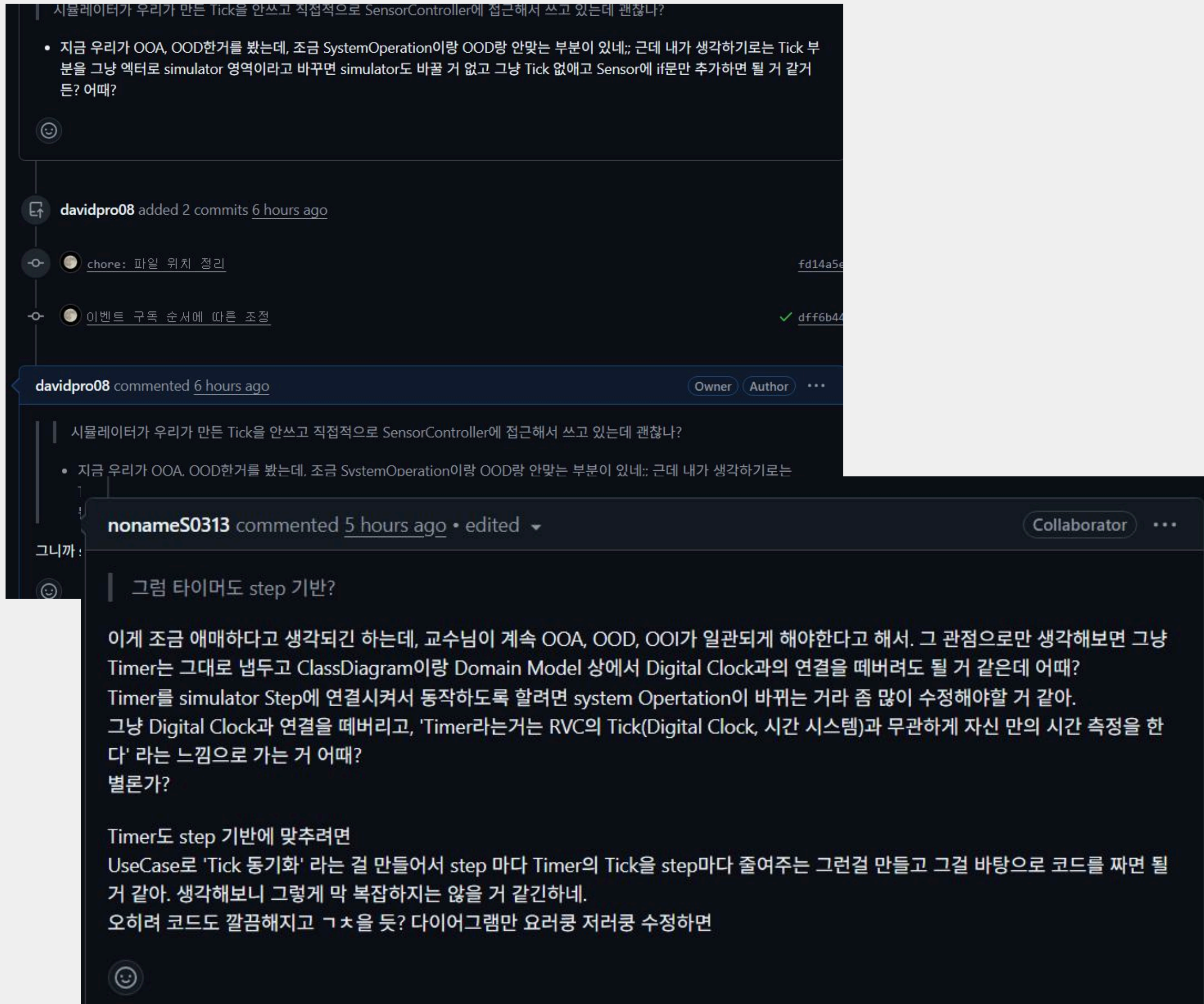
    // 행을 위에서부터 그릴 때 논리 y가 증가할수록 화면 위쪽에 오도록 반전
    for (int row = 0; row < height; ++row) {
        const int y = height - 1 - row;
        std::string line;
        line.reserve(width);
        for (int x = 0; x < width; ++x) {
            Point current(x, y);
            if (robot.isEqual(current)) {
                char robotSymbol = 'R';
                // 반전 후 화면 기준 위/아래에 맞춘 기호
                if (direction.x == 0 && direction.y == -1) robotSymbol = 'v';
                if (direction.x == 0 && direction.y == 1) robotSymbol = '^';
                if (direction.x == -1 && direction.y == 0) robotSymbol = '<';
                if (direction.x == 1 && direction.y == 0) robotSymbol = '>';
                line.push_back(robotSymbol);
                continue;
            }

            switch (cells[y][x]) {
                case CellType::Wall:
                    line.push_back('#');
```

- Code Nesting 문제: Sonarqube는 3회 이상의 Nesting 을 Maintainability의 심각한 오류로 잡음
- Simulator 상에서 화면에 그리는 메소드가 심각하게 Nesting 되어 있었음
- 여러 메소드로 잘개 쪼개어 가독성 + 유지보수성 확보

```
+     if (direction.x == -1 && direction.y == 0) return '<';
+     if (direction.x == 1 && direction.y == 0) return '>';
+     return 'R';
+ }
+
+ char GridMap::cellSymbol(CellType cell) {
+     switch (cell) {
+         case CellType::Wall:     return '#';
+         case CellType::Dust:     return '*';
+         case CellType::Cleaned: return 'x';
+         case CellType::Empty:
+         default:                 return '.';
+     }
+ }
+
+ std::string GridMap::renderRow(int y, Point robot, Point direction) const {
+     std::string line;
+     line.reserve(width);
+     for (int x = 0; x < width; ++x) {
+         Point current(x, y);
+         if (robot.isEqual(current)) {
+             line.push_back(robotSymbol(direction));
+         } else {
+             line.push_back(cellSymbol(cells[y][x]));
+         }
+     }
+ }
```

# Trouble Shooting 2



- Simulator 구현 중 Tick의 내/외부 영역에 대해서 토론
  - OOD랑 안 맞는 부분이 존재함
  - Tick을 내부로 판단함으로써 생긴 문제
  - 외부 Simulator에서 한번 과정을 수행하는 것(이동 + 먼지 흡입 + 장애물 처리 등등...) 을 Tick으로 규정하면 해결된다는 걸 알았음
- 조원들끼리 얘기하고 명세를 바꾸기로 결정
- 코드를 크게 바꾸지 않고 해결

[https://github.com/davidpro08/2026\\_1\\_OOAD/pull/14](https://github.com/davidpro08/2026_1_OOAD/pull/14)