

# 객체지향개발방법론

## 팀프로젝트 #4 - OOI - Implementation



**3팀**

202213351 김태성

202111382 최성준

202011380 최용근

202011434 최원탁

# Unit Test

# Google Test 결과

```
▶ ctest --test-dir build --output-on-failure
Test project /Users/seongjun/Documents/coding/Robot-Vacuum-Cleaner/build
  Start 1: UC01ControllerStartTest.OnTimerTickWhenClearKeepsCleaningForward
1/56 Test #1: UC01ControllerStartTest.OnTimerTickWhenClearKeepsCleaningForward ..... Passed 0.21 sec
  Start 2: ControllerSingletonTest.GetInstance_ReturnsSameInstance
2/56 Test #2: ControllerSingletonTest.GetInstance_ReturnsSameInstance ..... Passed 0.01 sec
  Start 3: ControllerTest.ReceiveCommand_Start_NoThrow
3/56 Test #3: ControllerTest.ReceiveCommand_Start_NoThrow ..... Passed 0.32 sec
  Start 4: ControllerTest.ReceiveCommand_Stop_NoThrow
4/56 Test #4: ControllerTest.ReceiveCommand_Stop_NoThrow ..... Passed 0.01 sec
  Start 5: ControllerTest.Sensors_ReadTrueFromTextGrid
5/56 Test #5: ControllerTest.Sensors_ReadTrueFromTextGrid ..... Passed 0.01 sec
  Start 6: ControllerTest.StartTimer_NoThrow
6/56 Test #6: ControllerTest.StartTimer_NoThrow ..... Passed 0.32 sec
  Start 7: ControllerTest.StopTimer_NoThrow
7/56 Test #7: ControllerTest.StopTimer_NoThrow ..... Passed 0.01 sec
  Start 8: ControllerTest.OnTimerTick_NoThrow
8/56 Test #8: ControllerTest.OnTimerTick_NoThrow ..... Passed 0.01 sec
  Start 9: ObstacleSensorHandlerTest.GetObstacleState_WhenAllClear_ReturnsClear
9/56 Test #9: ObstacleSensorHandlerTest.GetObstacleState_WhenAllClear_ReturnsClear ..... Passed 0.01 sec
  Start 10: ObstacleSensorHandlerTest.GetObstacleState_WhenOnlyFrontBlocked_ReturnsFrontBlocked
10/56 Test #10: ObstacleSensorHandlerTest.GetObstacleState_WhenOnlyFrontBlocked_ReturnsFrontBlocked ..... Passed 0.01 sec
  Start 11: ObstacleSensorHandlerTest.GetObstacleState_WhenFrontAndLeftBlocked_ReturnsFrontLeftBlocked
11/56 Test #11: ObstacleSensorHandlerTest.GetObstacleState_WhenFrontAndLeftBlocked_ReturnsFrontLeftBlocked ..... Passed 0.01 sec
  Start 12: ObstacleSensorHandlerTest.GetObstacleState_WhenFrontLeftRightBlocked_ReturnsCornerTrapped
12/56 Test #12: ObstacleSensorHandlerTest.GetObstacleState_WhenFrontLeftRightBlocked_ReturnsCornerTrapped ..... Passed 0.01 sec
  Start 13: ObstacleSensorHandlerTest.GetObstacleState_WhenAllBlocked_ReturnsAllBlocked
13/56 Test #13: ObstacleSensorHandlerTest.GetObstacleState_WhenAllBlocked_ReturnsAllBlocked ..... Passed 0.01 sec
  Start 14: ObstacleSensorHandlerTest.GetObstacleState_WhenLeftRightRearBlocked_ReturnsLeftRightRearBlocked
14/56 Test #14: ObstacleSensorHandlerTest.GetObstacleState_WhenLeftRightRearBlocked_ReturnsLeftRightRearBlocked ... Passed 0.01 sec
  Start 15: ObstacleSensorHandlerTest.GetObstacleState_WhenLeftAndRightBlocked_ReturnsLeftRightBlocked
15/56 Test #15: ObstacleSensorHandlerTest.GetObstacleState_WhenLeftAndRightBlocked_ReturnsLeftRightBlocked ..... Passed 0.00 sec
  Start 16: ObstacleSensorHandlerTest.GetObstacleState_WhenOnlyLeftOpened_ReturnsLeftOpened
16/56 Test #16: ObstacleSensorHandlerTest.GetObstacleState_WhenOnlyLeftOpened_ReturnsLeftOpened ..... Passed 0.00 sec
  Start 17: ObstacleSensorHandlerTest.GetObstacleState_WhenOnlyRightOpened_ReturnsRightOpened
17/56 Test #17: ObstacleSensorHandlerTest.GetObstacleState_WhenOnlyRightOpened_ReturnsRightOpened ..... Passed 0.00 sec
  Start 18: UseCase04Driver.ClearStateDoesNotStartCornerEscape
18/56 Test #18: UseCase04Driver.ClearStateDoesNotStartCornerEscape ..... Passed 0.00 sec
  Start 19: UseCase04Driver.FrontBlockedOnlyDoesNotStartUseCase04
19/56 Test #19: UseCase04Driver.FrontBlockedOnlyDoesNotStartUseCase04 ..... Passed 0.00 sec
  Start 20: UseCase04Driver.AllBlockedInitialStateDoesNotStartUseCase04
20/56 Test #20: UseCase04Driver.AllBlockedInitialStateDoesNotStartUseCase04 ..... Passed 0.00 sec
  Start 21: UseCase04Driver.CornetrappedStartsEscapeByStoppingCleanerAndMovingBackward
21/56 Test #21: UseCase04Driver.CornetrappedStartsEscapeByStoppingCleanerAndMovingBackward ..... Passed 0.00 sec
  Start 22: UseCase04Driver.RearBlockedAfterFirstReverseStopsMotor
22/56 Test #22: UseCase04Driver.RearBlockedAfterFirstReverseStopsMotor ..... Passed 0.00 sec
  Start 23: UseCase04Driver.RearStillBlockedOnNextTickKeepsMotorStopped
23/56 Test #23: UseCase04Driver.RearStillBlockedOnNextTickKeepsMotorStopped ..... Passed 0.00 sec
  Start 24: UseCase04Driver.RearClearedFromWaitPhaseDoesNotMoveUntilNextTick
24/56 Test #24: UseCase04Driver.RearClearedFromWaitPhaseDoesNotMoveUntilNextTick ..... Passed 0.00 sec
  Start 25: UseCase04Driver.SearchPhaseWithoutEscapeKeepsMovingBackward
25/56 Test #25: UseCase04Driver.SearchPhaseWithoutEscapeKeepsMovingBackward ..... Passed 0.00 sec
  Start 26: UseCase04Driver.SearchPhaseWithoutEscapeMovesBackwardOnEachTick
26/56 Test #26: UseCase04Driver.SearchPhaseWithoutEscapeMovesBackwardOnEachTick ..... Passed 0.00 sec
  Start 27: UseCase04Driver.SearchPhaseWithRearBlockedDoesNotFinishEscape
27/56 Test #27: UseCase04Driver.SearchPhaseWithRearBlockedDoesNotFinishEscape ..... Passed 0.00 sec
  Start 28: UseCase04Driver.SearchPhaseWithFrontLeftBlockedDoesNotFinishEscape
28/56 Test #28: UseCase04Driver.SearchPhaseWithFrontLeftBlockedDoesNotFinishEscape ..... Passed 0.00 sec
  Start 29: UseCase04Driver.SearchPhaseWithLeftOpenedTurnsLeftAndResumesCleaning
29/56 Test #29: UseCase04Driver.SearchPhaseWithLeftOpenedTurnsLeftAndResumesCleaning ..... Passed 0.00 sec
  Start 30: UseCase04Driver.SearchPhaseWithBothSidesOpenedChoosesLeftByPriority
30/56 Test #30: UseCase04Driver.SearchPhaseWithBothSidesOpenedChoosesLeftByPriority ..... Passed 0.00 sec
  Start 31: UseCase04Driver.SearchPhaseWithRightOpenedTurnsRightAndResumesCleaning
31/56 Test #31: UseCase04Driver.SearchPhaseWithRightOpenedTurnsRightAndResumesCleaning ..... Passed 0.00 sec
  Start 32: UseCase04Driver.WaitPhaseClearsThenRightOpenedFinishesOnFollowingTick
```

```
32/56 Test #32: UseCase04Driver.WaitPhaseClearsThenRightOpenedFinishesOnFollowingTick ..... Passed 0.00 sec
  Start 33: UseCase04Driver.CompletedEscapeNoLongerAdvancesOnNextClearTick
33/56 Test #33: UseCase04Driver.CompletedEscapeNoLongerAdvancesOnNextClearTick ..... Passed 0.00 sec
  Start 34: UseCase04Driver.EscapeCanStartAgainAfterPreviousEscapeCompleted
34/56 Test #34: UseCase04Driver.EscapeCanStartAgainAfterPreviousEscapeCompleted ..... Passed 0.00 sec
  Start 35: ControllerUC5Test.DustDetected_SetsIntensityHigh
35/56 Test #35: ControllerUC5Test.DustDetected_SetsIntensityHigh ..... Passed 0.00 sec
  Start 36: ControllerUC5Test.NoDust_NoCleanerCall
36/56 Test #36: ControllerUC5Test.NoDust_NoCleanerCall ..... Passed 0.00 sec
  Start 37: ControllerUC5Test.BoostTimeout_SetsIntensityNormal
37/56 Test #37: ControllerUC5Test.BoostTimeout_SetsIntensityNormal ..... Passed 3.01 sec
  Start 38: ControllerUC5Test.BeforeBoostTimeout_NormalNotCalled
38/56 Test #38: ControllerUC5Test.BeforeBoostTimeout_NormalNotCalled ..... Passed 0.01 sec
  Start 39: ControllerUC5Test.DustRedetectedDuringBoost_SetsHighAgain
39/56 Test #39: ControllerUC5Test.DustRedetectedDuringBoost_SetsHighAgain ..... Passed 0.01 sec
  Start 40: ControllerUC5Test.MultipleTicksNoDust_NoCleanerCall
40/56 Test #40: ControllerUC5Test.MultipleTicksNoDust_NoCleanerCall ..... Passed 0.01 sec
  Start 41: ControllerUC5Test.NotInIntensiveCleaning_TimeoutDoesNothing
41/56 Test #41: ControllerUC5Test.NotInIntensiveCleaning_TimeoutDoesNothing ..... Passed 0.01 sec
  Start 42: ControllerUC5Test.DustRedetected_ResetsBoostTimer
42/56 Test #42: ControllerUC5Test.DustRedetected_ResetsBoostTimer ..... Passed 2.12 sec
  Start 43: CleanerTest.StartCleaning_NoThrow
43/56 Test #43: CleanerTest.StartCleaning_NoThrow ..... Passed 0.01 sec
  Start 44: CleanerTest.StopCleaning_NoThrow
44/56 Test #44: CleanerTest.StopCleaning_NoThrow ..... Passed 0.01 sec
  Start 45: CleanerTest.StartThenStop_NoThrow
45/56 Test #45: CleanerTest.StartThenStop_NoThrow ..... Passed 0.01 sec
  Start 46: CleanerTest.SetIntensity_High_NoThrow
46/56 Test #46: CleanerTest.SetIntensity_High_NoThrow ..... Passed 0.01 sec
  Start 47: CleanerTest.SetIntensity_Normal_NoThrow
47/56 Test #47: CleanerTest.SetIntensity_Normal_NoThrow ..... Passed 0.01 sec
  Start 48: CleanerTest.SetIntensity_HighToNormal_NoThrow
48/56 Test #48: CleanerTest.SetIntensity_HighToNormal_NoThrow ..... Passed 0.01 sec
  Start 49: CleanerTest.SetIntensity_NormalToHigh_NoThrow
49/56 Test #49: CleanerTest.SetIntensity_NormalToHigh_NoThrow ..... Passed 0.01 sec
  Start 50: CleanerTest.StartCleaning_ThenSetIntensityHigh_NoThrow
50/56 Test #50: CleanerTest.StartCleaning_ThenSetIntensityHigh_NoThrow ..... Passed 0.01 sec
  Start 51: CleanerTest.StopCleaning_WithoutStart_NoThrow
51/56 Test #51: CleanerTest.StopCleaning_WithoutStart_NoThrow ..... Passed 0.00 sec
  Start 52: CleanerTest.SetIntensityHigh_PrintsCurrentIntensity
52/56 Test #52: CleanerTest.SetIntensityHigh_PrintsCurrentIntensity ..... Passed 0.01 sec
  Start 53: CleanerTest.SetIntensityNormal_PrintsCurrentIntensity
53/56 Test #53: CleanerTest.SetIntensityNormal_PrintsCurrentIntensity ..... Passed 0.00 sec
  Start 54: MotorTest.TurnLeftPrintsCurrentDirection
54/56 Test #54: MotorTest.TurnLeftPrintsCurrentDirection ..... Passed 0.00 sec
  Start 55: MotorTest.TurnRightPrintsCurrentDirection
55/56 Test #55: MotorTest.TurnRightPrintsCurrentDirection ..... Passed 0.00 sec
  Start 56: MotorTest.SequentialTurnsPrintUpdatedDirection
56/56 Test #56: MotorTest.SequentialTurnsPrintUpdatedDirection ..... Passed 0.00 sec

100% tests passed, 0 tests failed out of 56

Total Test time (real) = 6.31 sec
```

## Build 방법

cmake --build build

ctest --test-dir build --output-on-failure

# Google Test 결과

UC 번호	Test Case Name	Description	Condition	Expect	Pass/Fail
UC-01	<code>UC01ControllerStartTest.OnTimerTickWhenClearKeepsCleaningForward</code>	자동 청소 시작 후 장애물이 없을 때 청소와 전진이 유지되는지 검증	모든 <code>ObstacleSensorInterface</code> 가 <code>false</code> 를 반환하고 <code>CommandType::START</code> 를 입력	<code>CleanerInterface::startCleaning()</code> 과 <code>MotorInterface::moveForward()</code> 가 1회 이상 호출	Pass
UC-02	<code>ControllerSingletonTest.GetInstance_ReturnsSameInstance</code>	<code>Controller</code> singleton이 항상 동일한 인스턴스를 반환하는지 검증	<code>Controller::getInstance()</code> 를 두 번 호출	두 참조의 주소가 동일	Pass
UC-02	<code>ControllerTest.ReceiveCommand_Start_NoThrow</code>	시작 명령 처리 중 예외가 발생하지 않는지 검증	<code>CommandType::START</code> 를 별도 thread에서 실행 후 timer 정지	<code>receiveCommand()</code> 가 예외 없이 종료	Pass
UC-02	<code>ControllerTest.ReceiveCommand_Stop_NoThrow</code>	정지 명령 처리 중 예외가 발생하지 않는지 검증	<code>CommandType::STOP</code> 입력	<code>receiveCommand()</code> 가 예외 없이 종료	Pass
UC-02	<code>ControllerTest.Sensors_ReadTrueFromTextGrid</code>	<code>test.txt</code> 기반 센서 값이 정상적으로 읽히는지 검증	<code>TextGridReader</code> 로 생성한 전방, 좌측, 우측, 후방, 먼지 센서를 사용	모든 센서 읽기 결과가 <code>true</code>	Pass
UC-02	<code>ControllerTest.StartTimer_NoThrow</code>	timer loop 시작과 종료에 예외 없이 수행되는지 검증	별도 thread에서 <code>startTimer()</code> 실행 후 250ms 뒤 <code>stopTimer()</code> 호출	<code>startTimer()</code> 가 예외 없이 종료	Pass
UC-02	<code>ControllerTest.StopTimer_NoThrow</code>	timer 정지 호출이 안전한지 검증	<code>stopTimer()</code> 단독 호출	예외 없이 종료	Pass
UC-02	<code>ControllerTest.OnTimerTick_NoThrow</code>	단일 timer tick 처리 중 예외가 발생하지 않는지 검증	<code>Controller</code> fixture 생성 후 <code>onTimerTick()</code> 호출	예외 없이 종료	Pass
UC-02	<code>ObstacleSensorHandlerTest.GetObstacleState_WhenAllClear_ReturnsClear</code>	모든 방향이 열린 상태를 올바르게 분류하는지 검증	front=false, left=false, right=false, rear=false	<code>ObstacleState::CLEAR</code> 반환	Pass
UC-02	<code>ObstacleSensorHandlerTest.GetObstacleState_WhenOnlyFrontBlocked_ReturnsFrontBlocked</code>	전방만 막힌 상태를 올바르게 분류하는지 검증	front=true, left=false, right=false, rear=false	<code>ObstacleState::FRONT_BLOCKED</code> 반환	Pass
UC-02	<code>ObstacleSensorHandlerTest.GetObstacleState_WhenFrontAndLeftBlocked_ReturnsFrontLeftBlocked</code>	전방과 좌측이 막힌 상태를 올바르게 분류하는지 검증	front=true, left=true, right=false, rear=false	<code>ObstacleState::FRONT_LEFT_BLOCKED</code> 반환	Pass

# Google Test 결과

UC-04	<code>ObstacleSensorHandlerTest.GetObstacleState_whenFrontLeftRightBlocked_ReturnsCornerTrapped</code>	전방, 좌측, 우측이 막힌 corner trapped 진입 조건을 검증	front=true, left=true, right=true, rear=false	<code>ObstacleState::CORNER_TRAPPED</code> 반환	Pass
UC-04	<code>ObstacleSensorHandlerTest.GetObstacleState_whenAllBlocked_ReturnsAllBlocked</code>	모든 방향이 막힌 상태를 올바르게 분류하는지 검증	front=true, left=true, right=true, rear=true	<code>ObstacleState::ALL_BLOCKED</code> 반환	Pass
UC-04	<code>ObstacleSensorHandlerTest.GetObstacleState_whenLeftRightRearBlocked_ReturnsLeftRightRearBlocked</code>	좌측, 우측, 후방이 막힌 후진 대기 상태를 분류하는지 검증	front=false, left=true, right=true, rear=true	<code>ObstacleState::LEFT_RIGHT_REAR_BLOCKED</code> 반환	Pass
UC-04	<code>ObstacleSensorHandlerTest.GetObstacleState_whenLeftAndRightBlocked_ReturnsLeftRightBlocked</code>	좌우가 모두 막힌 상태를 올바르게 분류하는지 검증	front=false, left=true, right=true, rear=false	<code>ObstacleState::LEFT_RIGHT_BLOCKED</code> 반환	Pass
UC-04	<code>ObstacleSensorHandlerTest.GetObstacleState_whenOnlyLeftOpened_ReturnsLeftOpened</code>	왼쪽만 열린 탈출 가능 상태를 분류하는지 검증	front=true, left=false, right=true, rear=true	<code>ObstacleState::LEFT_OPENED</code> 반환	Pass
UC-04	<code>ObstacleSensorHandlerTest.GetObstacleState_whenOnlyRightOpened_ReturnsRightOpened</code>	오른쪽만 열린 탈출 가능 상태를 분류하는지 검증	front=true, left=true, right=false, rear=true	<code>ObstacleState::RIGHT_OPENED</code> 반환	Pass
UC-04	<code>UseCase04Driver.ClearStateDoesNotStartCornerEscape</code>	장애물이 없는 상태에서 corner escape가 시작되지 않는지 검증	front=false, left=false, right=false, rear=true	<code>moveForward</code> 호출, 청소 시작 1회, 청소 정지 0회	Pass
UC-04	<code>UseCase04Driver.FrontBlockedOnlyDoesNotStartUseCase04</code>	단순 전방 장애물이 UC-04가 아닌 UC-03 회피로 처리되는지 검증	front=true, left=false, right=false, rear=false	<code>stop</code> , <code>turnDirection</code> , <code>moveForward</code> 순서 호출, 방향은 <code>Direction::LEFT</code>	Pass
UC-04	<code>UseCase04Driver.AllBlockedInitialStateDoesNotStartUseCase04</code>	초기 <code>ALL_BLOCKED</code> 상태에서 UC-04 탈출 절차가 시작되지 않는지 검증	front=true, left=true, right=true, rear=true	motor 호출 없음, 청소 정지 0회	Pass
UC-04	<code>UseCase04Driver.CornetrappedStartsEscapeByStoppingCleanerAndMovingBackward</code>	<code>CORNER_TRAPPED</code> 감지 시 청소를 멈추고 1차 후진을 시작하는지 검증	front=true, left=true, right=true, rear=false	청소 정지 1회, <code>moveBackward</code> 1회 호출	Pass

# Google Test 결과

UC-04	<code>UseCase04Driver.RearBlockedAfterFirstReverseStopsMotor</code>	1차 후진 직후 후방 장애물이 감지되면 motor를 정지하는지 검증	처음 rear=false, 다음 rear=true인 센서 sequence	<code>moveBackward</code> 후 <code>stop</code> 호출	Pass
UC-04	<code>UseCase04Driver.RearStillBlockedOnNextTickKeepsMotorStopped</code>	후방 장애물이 계속 감지되면 다음 tick에서도 정지 상태를 유지하는지 검증	후방 대기 단계 진입 후 front=false, left=true, right=true, rear=true	<code>stop</code> 1회 호출, 청소 재시작 없음	Pass
UC-04	<code>UseCase04Driver.RearClearedFromWaitPhaseDoesNotMoveUntilNextTick</code>	후방 대기 단계에서 후방이 열리면 탐색 단계로만 전이되는지 검증	후방 대기 단계에서 front=false, left=true, right=true, rear=false	즉시 motor 호출 없음, 청소 재시작 없음	Pass
UC-04	<code>UseCase04Driver.SearchPhaseWithoutEscapeKeepsMovingBackward</code>	탈출 방향이 없으면 후진을 유지하는지 검증	탐색 단계에서 front=false, left=true, right=true, rear=false	<code>moveBackward</code> 1회 호출, 청소 재시작 없음	Pass
UC-04	<code>UseCase04Driver.SearchPhaseWithoutEscapeMovesBackwardOnEachTick</code>	탈출 방향이 계속 없으면 tick마다 후진하는지 검증	탐색 단계에서 같은 blocked 상태로 <code>onTimerTick()</code> 2회 호출	<code>moveBackward</code> 가 2회 연속 호출	Pass
UC-04	<code>UseCase04Driver.SearchPhaseWithRearBlockedDoesNotFinishEscape</code>	탐색 중 후방까지 막힌 상태를 탈출 완료로 처리하지 않는지 검증	탐색 단계에서 front=false, left=true, right=true, rear=true	<code>moveBackward</code> 만 호출, 회전 없음, 청소 재시작 없음	Pass
UC-04	<code>UseCase04Driver.SearchPhaseWithFrontLeftBlockedDoesNotFinishEscape</code>	탐색 중 UC-03성 장애물 상태를 UC-04 완료로 처리하지 않는지 검증	탐색 단계에서 front=true, left=true, right=false, rear=false	<code>moveBackward</code> 만 호출, 회전 없음, 청소 재시작 없음	Pass
UC-04	<code>UseCase04Driver.SearchPhaseWithLeftOpenedTurnsLeftAndResumesCleaning</code>	왼쪽 탈출 가능 상태에서 좌회전 후 청소를 재개하는지 검증	탐색 단계에서 front=false, left=false, right=true, rear=false	<code>moveBackward</code> , <code>turnDirection</code> , <code>moveForward</code> 순서 호출, 방향은 <code>Direction::LEFT</code> , 청소 시작 1회	Pass
UC-04	<code>UseCase04Driver.SearchPhaseWithBothSidesOpenedChoosesLeftByPriority</code>	양쪽 탈출 가능 시 왼쪽 우선 정책을 적용하는지 검증	탐색 단계에서 front=false, left=false, right=false, rear=false	회전 방향이 <code>Direction::LEFT</code> , 청소 시작 1회	Pass
UC-04	<code>UseCase04Driver.SearchPhaseWithRightOpenedTurnsRightAndResumesCleaning</code>	오른쪽만 탈출 가능할 때 우회전 후 청소를 재개하는지 검증	탐색 단계에서 front=false, left=true, right=false, rear=false	<code>moveBackward</code> , <code>turnDirection</code> , <code>moveForward</code> 순서 호출, 방향은 <code>Direction::RIGHT</code> , 청소 시작 1회	Pass

# Google Test 결과

UC-04	<code>UseCase04Driver.WaitPhaseClearsThenRightOpenedFinishesOnFollowingTick</code>	후방 대기 해제 후 다음 tick에서 오른쪽 탈출을 완료하는지 검증	rear blocked로 대기 진입 후 rear clear, 다음 tick에서 right opened	중간 tick에는 motor 호출 없음, 다음 tick에서 <code>Direction::RIGHT</code> 회전과 청소 시작	Pass
UC-04	<code>UseCase04Driver.CompletedEscapeNoLongerAdvancesOnNextClearTick</code>	탈출 완료 후 다음 clear tick에서 UC-04 내부 단계가 계속 실행되지 않는지 검증	오른쪽 탈출 완료 후 front=false, left=false, right=false, rear=true	일반 주행으로 <code>moveForward</code> 호출, 청소 시작 1회	Pass
UC-04	<code>UseCase04Driver.EscapeCanStartAgainAfterPreviousEscapeCompleted</code>	한 번 탈출 완료 후 다시 <code>CORNER_TRAPPED</code> 가 감지되면 UC-04를 재시작하는지 검증	이전 탈출 완료 후 front=true, left=true, right=true, rear=false	청소 정지 1회, <code>moveBackward</code> 1회 호출	Pass
UC-05	<code>ControllerUC5Test.DustDetected_SetsIntensityHigh</code>	먼지 감지 시 청소 강도를 높이는지 검증	<code>DustSensor::isDustDetected()</code> 가 <code>true</code> 반환	첫 cleaner 호출이 <code>setIntensity(HIGH)</code>	Pass
UC-05	<code>ControllerUC5Test.NoDust_NoCleanerCall</code>	먼지가 없을 때 cleaner 명령이 발생하지 않는지 검증	<code>DustSensor::isDustDetected()</code> 가 <code>false</code> 반환, obstacle sensor 없음	cleaner 호출 없음	Pass
UC-05	<code>ControllerUC5Test.BoostTimeout_SetsIntensityNormal</code>	boost cleaning 시간이 만료되면 일반 강도로 복귀하는지 검증	먼지 감지 후 3초 대기, 이후 먼지 미감지	<code>setIntensity(NORMAL)</code> 호출	Pass
UC-05	<code>ControllerUC5Test.BeforeBoostTimeout_NormalNotCalled</code>	boost timeout 전에는 일반 강도 복귀가 발생하지 않는지 검증	먼지 감지 직후 다음 tick에서 먼지 미감지	<code>setIntensity(NORMAL)</code> 호출 없음	Pass
UC-05	<code>ControllerUC5Test.DustRedetectedDuringBoost_SetsHighAgain</code>	boost cleaning 중 먼지가 재감지되면 high 강도 명령을 다시 내리는지 검증	먼지 감지 상태로 <code>onTimerTick()</code> 2회 호출	<code>setIntensity(HIGH)</code> 가 2회 호출	Pass
UC-05	<code>ControllerUC5Test.MultipleTicksNoDust_NoCleanerCall</code>	여러 tick 동안 먼지가 없으면 cleaner 명령이 발생하지 않는지 검증	먼지 미감지 상태로 <code>onTimerTick()</code> 5회 호출	cleaner 호출 없음	Pass

# Google Test 결과

UC-05	<code>ControllerUC5Test.NotInIntensiveCleaning_TimeoutDoesNothing</code>	intensive cleaning 상태가 아닐 때 timeout 체크가 영향을 주지 않는지 검증	먼지 미감지 상태에서 단일 tick 실행	cleaner 호출 없음	Pass
UC-05	<code>ControllerUC5Test.DustRedetected_ResetsBoostTimer</code>	먼지 재감지 시 boost timer 가 reset되어 원래 만료 시점에 normal 복귀하지 않는지 검증	먼지 감지 후 1초 뒤 재감지, 1.1초 뒤 먼지 미감지	<code>setIntensity(NORMAL)</code> 호출 없음	Pass
UC-05/ 공통	<code>CleanerTest.StartCleaning_NoThrow</code>	청소 시작 동작이 예외 없이 수행되는지 검증	<code>Cleaner</code> 생성 후 <code>startCleaning()</code> 호출	예외 없음	Pass
UC-05/ 공통	<code>CleanerTest.StopCleaning_NoThrow</code>	청소 정지 동작이 예외 없이 수행되는지 검증	<code>Cleaner</code> 생성 후 <code>stopCleaning()</code> 호출	예외 없음	Pass
UC-05/ 공통	<code>CleanerTest.StartThenStop_NoThrow</code>	청소 시작 후 정지 흐름이 안전한지 검증	<code>startCleaning()</code> 호출 후 <code>stopCleaning()</code> 호출	두 호출 모두 예외 없음	Pass
UC-05	<code>CleanerTest.SetIntensity_High_NoThrow</code>	청소 강도를 high로 변경할 때 예외가 없는지 검증	<code>CleaningIntensity::HIGH</code> 입력	예외 없음	Pass
UC-05	<code>CleanerTest.SetIntensity_Normal_NoThrow</code>	청소 강도를 normal로 변경할 때 예외가 없는지 검증	<code>CleaningIntensity::NORMAL</code> 입력	예외 없음	Pass
UC-05	<code>CleanerTest.SetIntensity_HighToNormal_NoThrow</code>	high에서 normal로 강도 전환이 안전한지 검증	<code>HIGH</code> 설정 후 <code>NORMAL</code> 설정	두 호출 모두 예외 없음	Pass
UC-05	<code>CleanerTest.SetIntensity_NormalToHigh_NoThrow</code>	normal에서 high로 강도 전환이 안전한지 검증	<code>NORMAL</code> 설정 후 <code>HIGH</code> 설정	두 호출 모두 예외 없음	Pass
UC-05	<code>CleanerTest.StartCleaning_ThenSetIntensityHigh_NoThrow</code>	청소 시작 후 high 강도 설정이 안전한지 검증	<code>startCleaning()</code> 호출 후 <code>setIntensity(HIGH)</code> 호출	두 호출 모두 예외 없음	Pass
UC-05/ 공통	<code>CleanerTest.StopCleaning_WithoutStart_NoThrow</code>	청소 시작 없이 정지 호출을 해도 안전한지 검증	새 <code>Cleaner</code> 에서 <code>stopCleaning()</code> 호출	예외 없음	Pass
UC-05	<code>CleanerTest.SetIntensityHigh_PrintsCurrentIntensity</code>	high 강도 설정 시 현재 강도 출력이 맞는지 검증	stdout capture 후 <code>setIntensity(HIGH)</code> 호출	출력이 <code>[Cleaner] intensity: HIGH\n</code> 와 동일	Pass
UC-05	<code>CleanerTest.SetIntensityNormal_PrintsCurrentIntensity</code>	normal 강도 설정 시 현재 강도 출력이 맞는지 검증	stdout capture 후 <code>setIntensity(NORMAL)</code> 호출	출력이 <code>[Cleaner] intensity: NORMAL\n</code> 와 동일	Pass

# Code Review



myassa2000 reviewed [4 hours ago](#)

[View reviewed changes](#)

include/Controller.h

```
15 15 Instant boostEndTime;
16 + bool timerActive_ = false;
17 + CleanerInterface& cleaner_;
18 + MotorInterface& motor_;
```



myassa2000 [4 hours ago](#)

...

성준님은  
CleanerInterface\* cleanerInterface  
MotorInterface\* motorInterface  
이렇게 하셔서 통일해야 할 것 같습니다.



RoundTable02 [3 hours ago](#)

...

필수 의존성이라 &가 오히려 null 불가를 표현해서 더 나은 것 같은데, 어떠신가요? [@seongjun-choi](#)



seongjun-choi [3 hours ago](#)

...

&로 통일하겠습니다.



Reply...

Resolve conversation

tests/uc01\_controller\_start\_test.cpp

```
29 + }
30 + };
31 +
32 + class FakeMotor : public MotorInterface
```



myassa2000 [4 hours ago](#)

...

이 클래스 상속 받는 인터페이스(MotorInterface.h)에 virtual ~MotorInterface() = default;가 필요할 것 같습니다.



RoundTable02 [3 hours ago](#)

...

넵 그대로 수정 진행하겠습니다.



# Code Review

include/Controller.h

```
29 41 void stopTimer();
30 42 void onTimerTick();
43 + SystemState getSystemState() const;
44 + bool isTimerActive() const;
```

 **myassa2000** 4 hours ago ...

getSystemState()와 isTimerActive()는 getter 아닌가요?? 혹시 맞다면 getter는 쓰면 안 되는 것 아닌가요??



 **RoundTable02** 3 hours ago ...

네 getter는 지양하는 편이 좋을 것 같습니다. receiveCommand의 return으로 상태를 받아서 검증하도록 수정해보겠습니다



 **myassa2000** 3 hours ago ...

네 확인했습니다!!



 **RoundTable02** 3 hours ago ...

return 검증 방식으로 변경하면 수정이 많이 전파되는 문제가 있네요ㅠ 일단 state에 대해서는 예외적으로 getter 허용하는 방식으로 가면 어떨까요?



 **myassa2000** 3 hours ago ...

getSystemState()는 Stub의 호출 기록(cleaner.startCount, motor.moveForwardCount)으로 대체할 수 있어서 return 방식으로 바꾸지 않아도 됩니다!! isTimerActive()는 대체가 어려우니 해당 부분만 예외적으로 허용하는 건 어떨까요?? 이래도 수정이 많아진다면 state에 대해서는 예외적으로 getter 허용하는 방식으로 가야 할 것 같습니다!!



 **ts288** reviewed 4 hours ago

[View reviewed changes](#)

include/RearSensor.h Outdated

Comment on lines 4 to 11

```
4 4 class RearSensor
5 5 {
6 6 public:
7 - bool isBlocked();
7 + virtual ~RearSensor() = default;
8 + virtual bool isBlocked();
8 9 };
9 10
10 11 #endif
```

 **ts288** 4 hours ago ...

지금 작성해주신 방식은 RightSensor를 인터페이스처럼 두고 구현체에서 isBlocked() 를 재정의하려는 구조로 생각했습니다. 혹시 제가 생각한 게 맞을까요?

저도 같은 고민을 했었는데, 혹시 공통 인터페이스를 두는 건 어떨까요? 전후좌우 각 센서가 모두 동일하게 장애물 감지 역할을 수행하기 때문에, 각 방향별 인터페이스를 따로 두기보다 ObstacleSensor 같은 공통 인터페이스를 두고 구현하는 방향으로 수정해보았습니다! UC-02 방식 검토 부탁드립니다!



 **seongjun-choi** 3 hours ago Author ...

말씀해주신 것처럼 전후좌우 센서가 모두 동일하게 장애물 감지 역할을 하니 방향별 타입을 인터페이스처럼 쓰기보다는 ObstacleSensorInterface 같은 공통 인터페이스를 두는 UC-02 방식이 더 깔끔해 보입니다.

# Code Review

src/Controller.cpp **Outdated**

Comment on lines +110 to +115

```
110 + bool Controller::canRunCornerEscape() const
111 + {
112 +     return cleanerInterface != nullptr &&
113 +         motorInterface != nullptr &&
114 +         obstacleSensorHandler != nullptr;
115 + }
```

**ts288** 4 hours ago ...

혹시 인터페이스나 핸들러 주입을 참조 주입 방식으로 변경하는 건 어떨까요?  
체크해주시는 의존성이 모두 Controller 에 필수적인 의존성들이라, 참조 주입으로 변경하는 게 적절하다는 생각이 들어서요!  
참조 주입으로 변경할 경우에는 nullptr 상태가 방지되니까 체크 함수를 따로 만들 수고도 줄일 수 있을 것 같습니다!

참조 주입으로 변경하는 경우에는 -> 를 . 으로 변경해야 할 것 같습니다!

**seongjun-choi** 3 hours ago Author ...

참조 주입으로 변경하는 방식으로 수정하겠습니다!

**ts288** reviewed 4 hours ago View reviewed changes

src/Controller.cpp **Outdated**

```
24 44 bool Controller::pollDustSensor()
25 45 {
26 - return false;
46 + return dustSensor_>isDustDetected();
```

**ts288** 4 hours ago ...

UC-01, 02, 03 작성할 때에는 참조 주입 방식 사용했던 거 같은데, 주입 방식을 어떤 걸로 통일하는 게 좋을까요?

**myassa2000** 3 hours ago Author ...

네 확인했습니다!! 참조 주입 방식 사용해서 수정하겠습니다!!

**RoundTable02** reviewed 4 hours ago View reviewed changes

tests/CleanerTest.cpp Hide resolved

**RoundTable02** 4 hours ago ...

외부적으로 어떤 예외를 던지는지를 테스트 코드로 잘 작성하신 것 같습니다!  
그런데 내부 상태 변경이나, 다른 클래스와의 상호작용에 대한 테스트가 부족한 것 같아서 mock을 이용한 단위테스트를 더 추가해보면 어떨까요?

**myassa2000** 3 hours ago Author ...

cleaner가 다른 클래스에 의존하지 않는 클래스라서 mock을 쓸 대상이 없는데 혹시 어떻게 추가하면 좋을까요??  
ControllerTest.cpp은 Stub으로 상호작용 검증했습니다!!

**RoundTable02** 3 hours ago ...

앗 넵 테스트 코드 작성하신 거 확인했는데 충분한 것 같습니다!

# System Test

```
=====  
Robot Vacuum Cleaner - System Tests  
=====  
PASS: P01: START → cleaningStarted  
PASS: P02: STOP → stop  
PASS: P03: START response = 'OK START'  
PASS: P04: STOP response = 'OK STOP'  
PASS: P05: 클리어 경로 → unknown case 없음  
PASS: P06: 먼지 감지 → intensity HIGH  
PASS: P07: 부스트 타이머 만료 → intensity NORMAL  
PASS: P08: 앞 장애물 → direction LEFT  
PASS: P09: 앞+왼 장애물 → direction RIGHT  
PASS: P10: 코너 탈출 → direction LEFT  
PASS: N01: 'HELLO' → ERROR unsupported command  
PASS: N02: 'start' → ERROR unsupported command  
PASS: N03: 'stop' → ERROR unsupported command  
PASS: N04: 'STOPP' → ERROR unsupported command  
PASS: N05: 'STARTT' → ERROR unsupported command  
PASS: N06: 'START ' → ERROR unsupported command  
PASS: N07: ' START' → ERROR unsupported command  
PASS: N08: 빈 명령 → ERROR unsupported command  
PASS: N09: 중복 START → rejectStart  
PASS: N10: START 없이 STOP → OK STOP  
PASS: N10: START 없이 STOP → stop 로그  
PASS: N11: STOP 후 START → rejectStart  
PASS: N12: ALL_BLOCKED → unknown case  
PASS: N13: LEFT_RIGHT_BLOCKED → unknown case  
PASS: N14: LEFT_RIGHT_REAR_BLOCKED → unknown case  
PASS: N15: 앞+먼지 → 장애물 우선, direction LEFT  
PASS: N16: 복수 잘못된 명령 → 모두 ERROR  
PASS: N17: INTENSIVE_CLEANING 중 STOP → stop  
PASS: N18: '123' → ERROR unsupported command  
PASS: N19: 150자 명령 → ERROR unsupported command  
PASS: N20: 연속 STOP → 모두 OK STOP  
=====  
PASS: 31 FAIL: 0 TOTAL: 31  
=====
```

**감사합니다!**