소프트웨어 모델링

RE-ason Elevator

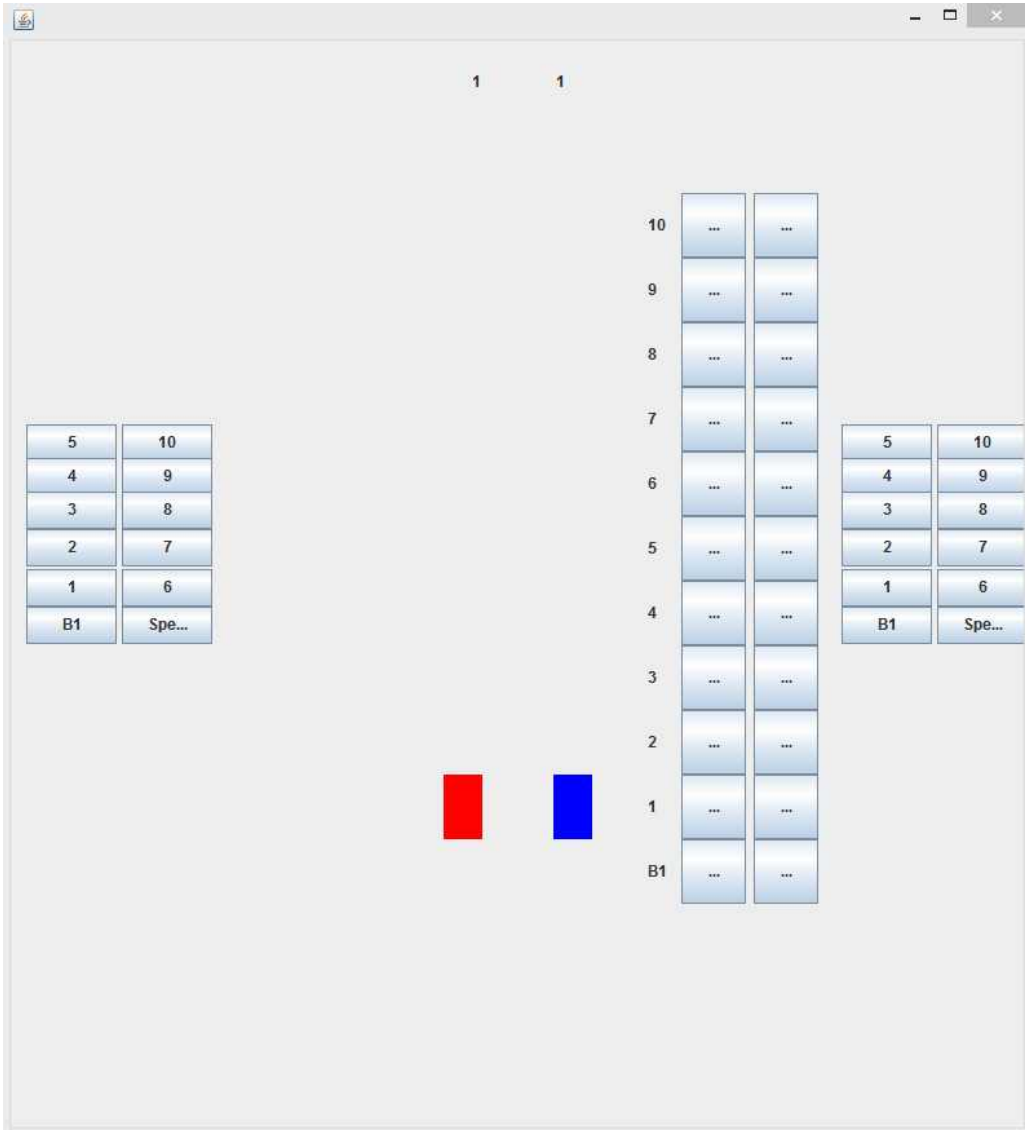
OOP 1st Cycle & Unit Test

The Team 1

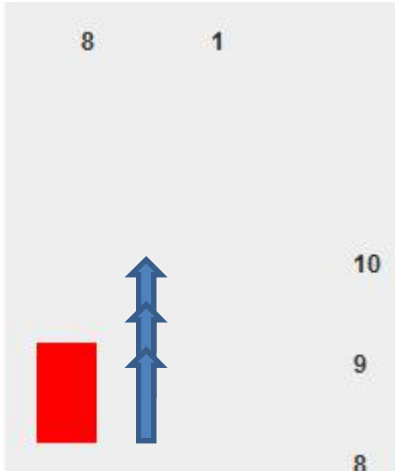
박현규 200911393
김도현 201211328
송해찬 201211358

DEMO

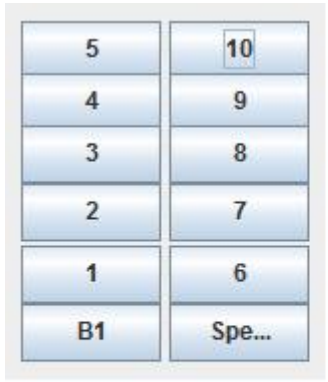
DEMO - 메인화면

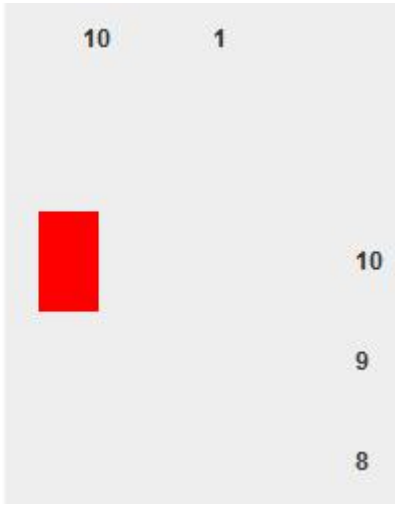


DEMO – 층 선택 버튼

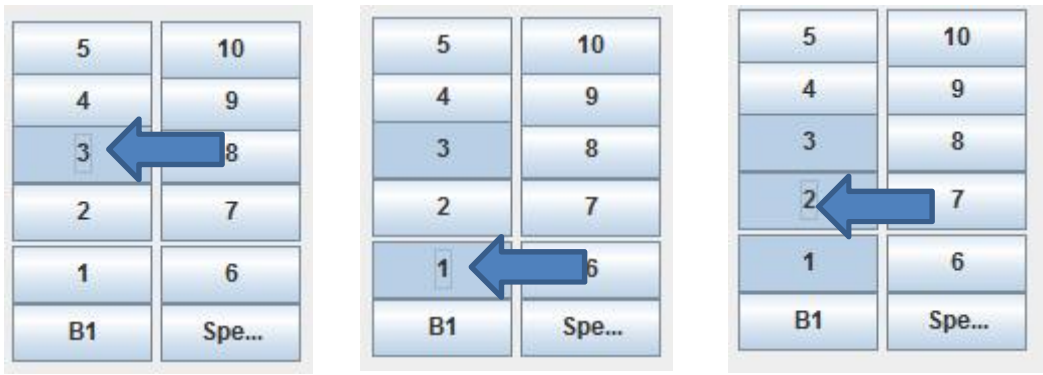


8
8
8
8
8
8
8
10
10
10
10
10
10
10





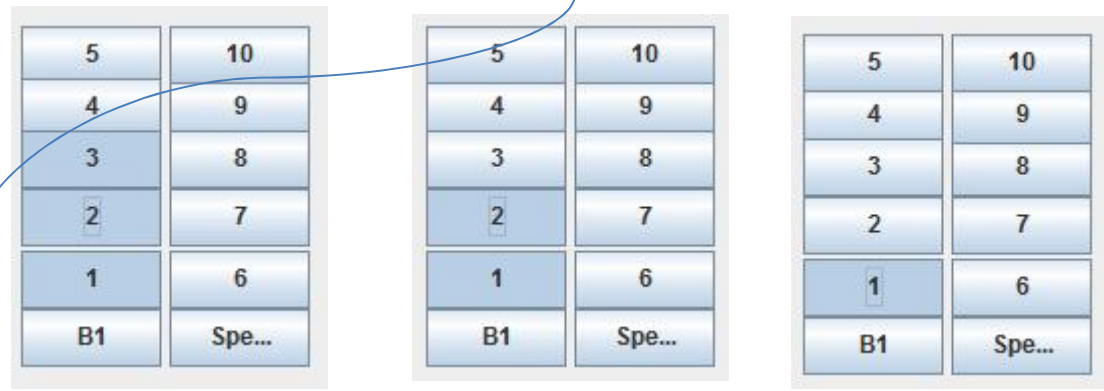
DEMO – 층 선택 & 이동



우선순위큐

층 선택
버튼 입력

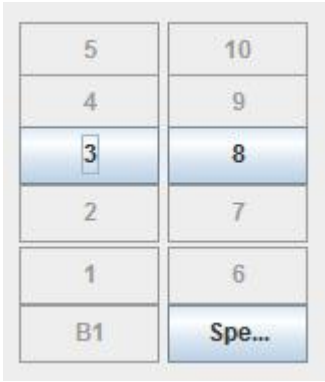
3 → 1, 3 → 1, 2, 3



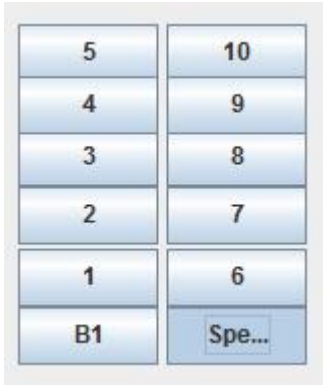
엘리베이터 이동

1, 2, 3 → 1, 2 → 1

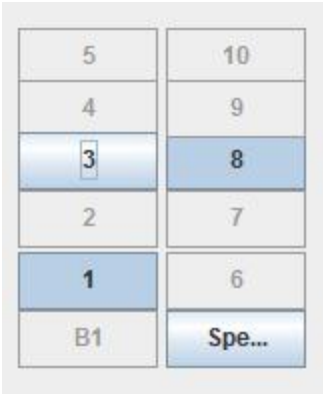
DEMO – 회원 버튼



비활성화 된 버튼들

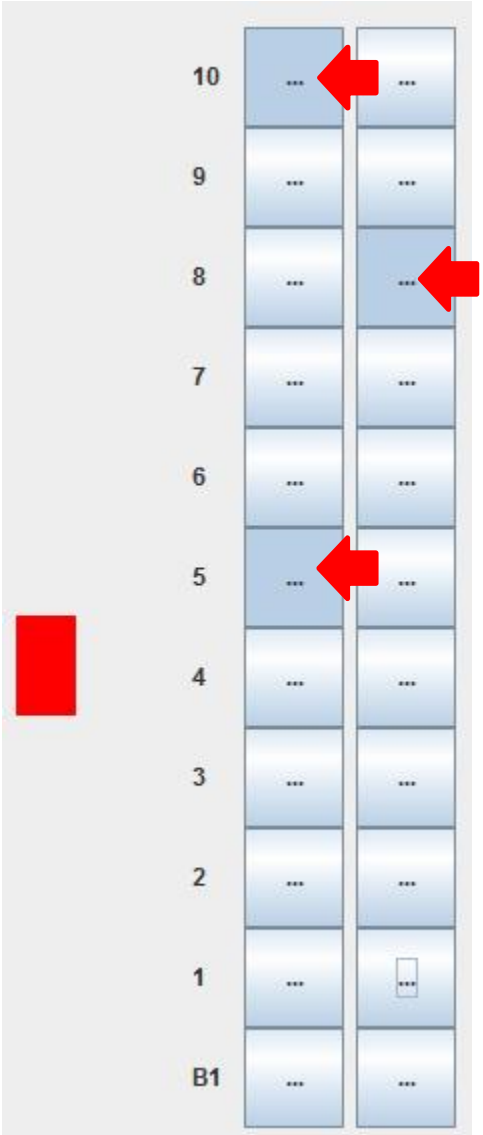


회원 기능으로
버튼을 활성화 시킴



비활성화 된 층
이동가능

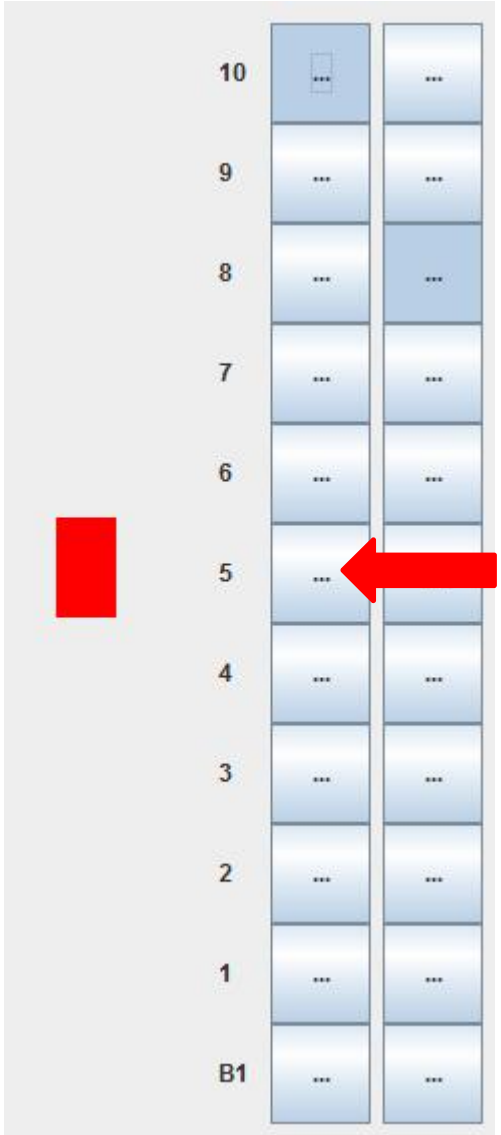
DEMO – 각 층 위 아래 버튼



위 아래 버튼으로

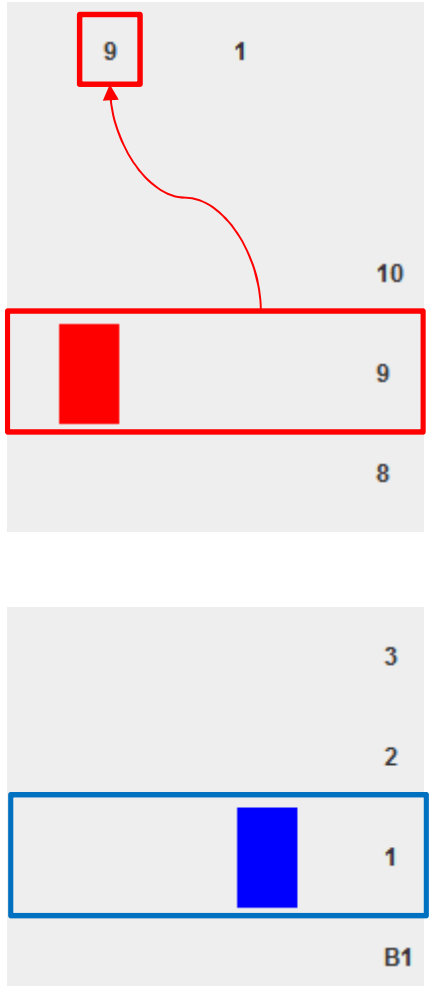
각 층에서

엘리베이터를
부른다



층에 도착하면
버튼이 비활성화

DEMO – 층 현황 디스플레이



UNIT TEST

UNIT TEST Order

제외항목

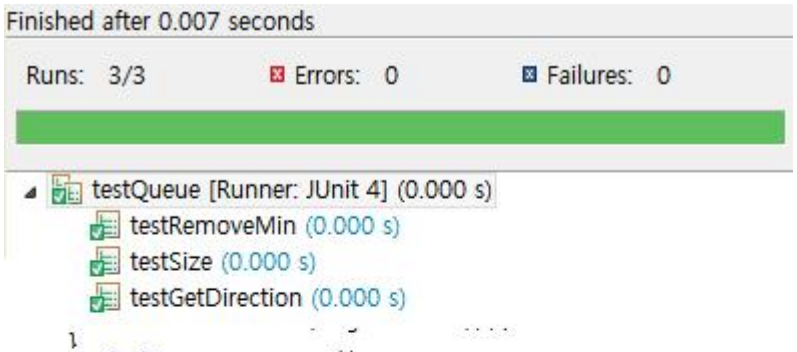
- Listener
- About GUI...
- Class 변수

검사항목

- Queue
- Movement Algorithm
- State
- Direction

Queue

Direction




```
queue left = new queue();

@Test
public void testSize() {
    assertEquals(left.size(), 1);
}

@Test
public void testRemoveMin() {
    left.insert(5);
    assertNotNull(left.removeMin());
}

@Test
public void testGetDirection() {
    assertNotNull(left.getDirection());
}
```

size()

a. 초기 queue 사이즈 == 1
b. Pressed button == queue.insert
c. queue 사이즈 증가

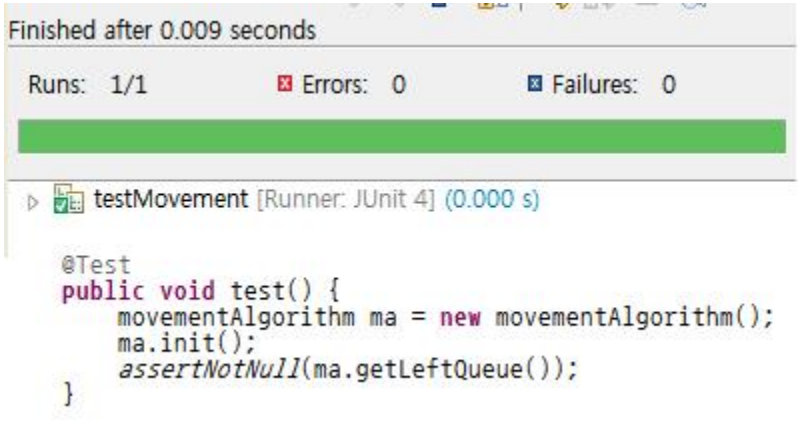
removeMin()

a. 엘리베이터 목표 층에 도착
b. queue에서 빠져나감
c. 도착한 층의 index 반환

getDirection()

a. 엘리베이터 방향 == 큐의 방향
b. 위/아래 방향 반환

Movement Algorithm



getLeftQueue()

a. movementAlgorithm 안에서 queue 생성
b. queue 반환

State

getState()

a. 엘리베이터 object의 현재 위치 계산

b. 현재 위치 반환