**<팀 최현호>**

**<Add-on>**

**설계 산출물**

|  |  |
| --- | --- |
| 문서버전 | 1.6 |
| 문서 ID | DO-2019-11-07 |
| 최종변경일 | 2019-12-02 |
| 문서상태 | 릴리즈 |

**요 약**

Sim과 함께 사용할 Add-on 시스템의 설계 산출물을 기술.

서브 시스템의 구성과 각 서브 시스템의 구조를 기술.

주요 산출물

* 아키텍처도
* 클래스도
* 교류도
* 상태도

**표 1 문서 변경 기록**

|  |  |  |  |
| --- | --- | --- | --- |
| 문서 이름 | <팀 최현호><Add-on> 설계 산출물 | | |
| 문서 ID | 현호-2002-001 | | |
| 1 | 0 | 2019-11-04 | 아키텍처 구성 |
| 1 | 2019-11-04 | 메인 클래스도 작성 |
| 2 | 2019-11-04 | 메인 클래스도 수정 |
| 3 | 2019-11-06 | Start 교류도 작성 |
| 4 | 2019-11-07 | Start 교류도 수정 |
| 5 | 2019-11-28 | 메인 클래스도 수정 |
| 6 | 2019-12-02 | 교류도 수정 |

**1 개 요**

* 1. **목 적**

본 문서는 “Add-on” 시스템의 설계 산출물을 기술한다.

* 전체 시스템이 어떤 서브 시스템으로 구성되어 있는지 “아키텍처도”를 통해 보여준다.
* 서브 시스템이 어떤 클래스로 구성되어 있고 클래스들간에 어떤 관계가 있는지를 “클래스도”를 통해서 보여준다.
* 객체간에 발생하는 동적인 행위를 “교류도”를 통해서 보여준다.
* 한 객체의 상태 변화를 “상태도”를 통해서 보여준다.
  1. **참고 문헌**

없음.

**2 아키텍처도**



전체 시스템은 아래의 서브 시스템으로 구성된다.

* UI 시스템
* DataManagement 시스템
* Data 시스템

UI 시스템

* 유저에게 UI를 보여주는 boundary 클래스들을 담고 있다. DataManagement 시스템이 데이터를 조작하는 명령을 수행하도록 요청하고 필요한 값들을 넘겨주는 역할을 한다.
* C# 언어를 사용하여 작성된다.

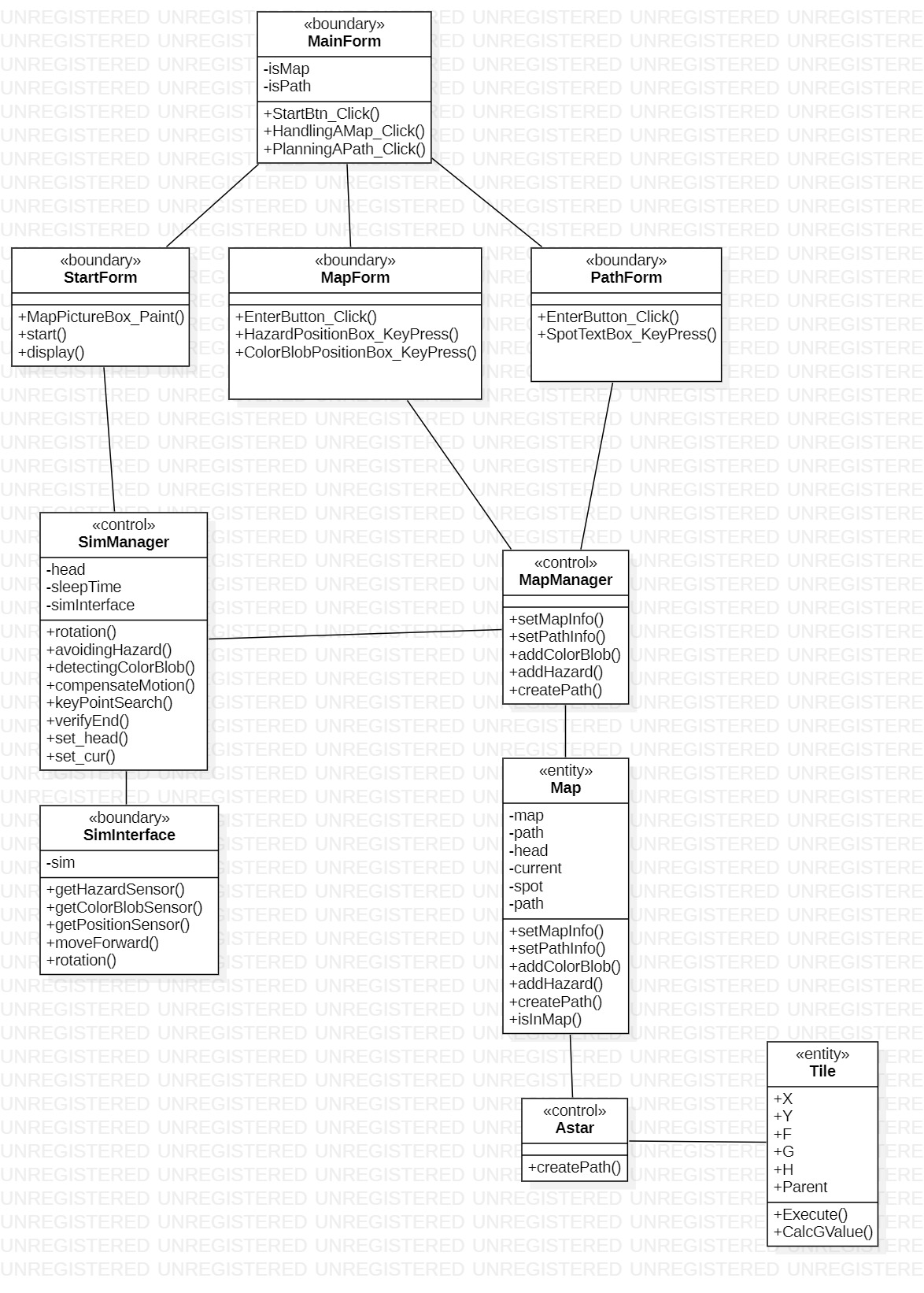
DataManagement 시스템

* UI 시스템에서 받아온 값들을 Data 시스템에 넘겨줘서 저장하도록 하는 역할을 한다.
* C# 언어를 사용하여 작성된다.

Data 시스템

* DataManagement 시스템에서 값을 받아, 데이터로 저장하는 역할을 한다.
* C# 언어를 사용하여 작성된다.

**3 메인 클래스도**



Display

* UI를 출력한다

setMapInfo

* 오퍼레이터로부터 맵 정보를 입력 받아 시스템에 저장한다.

setPathInfo

* 오퍼레이터로부터 경로 정보를 입력 받아 시스템에 저장한다.

createPath

* 입력 받은 경로를 이용해 경로를 생성한다.

keyPointSearch

* 경로를 따라 마지막까지 이동시키는 일을 한다.

avoidingHazard

* 위험지역 센서를 통해 위험지역을 피해가는 기능을 제공한다.

detectingColorBlob

* Color blob 센서를 통해 발견한 경우 지도에 추가하는 기능을 제공한다.

compensateMotion

* 로봇이 오작동한 경우 경로를 재설정해주는 기능을 제공한다.

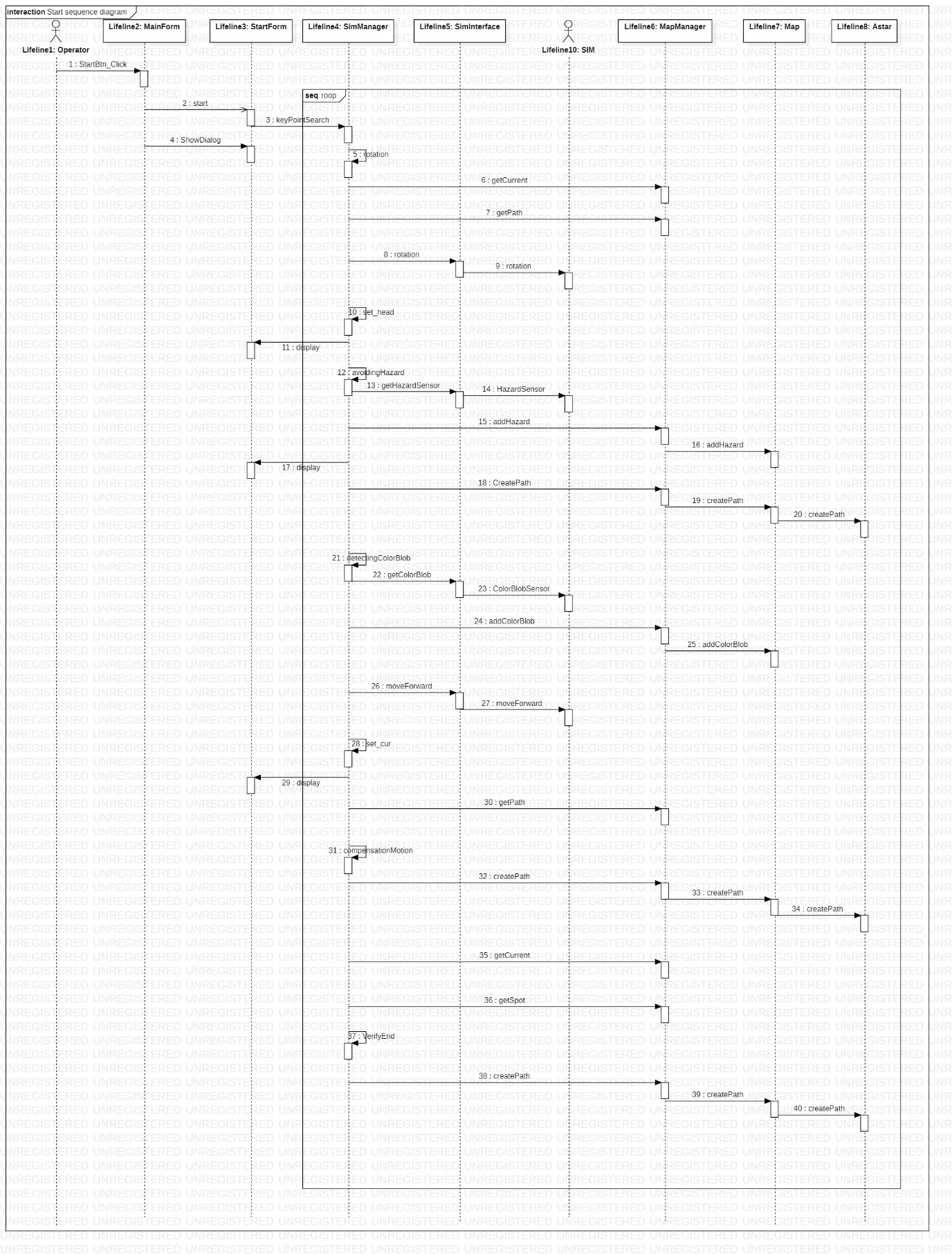
Rotation

* 로봇을 시계방향으로 90도 돌려주는 기능을 한다.

moveForward

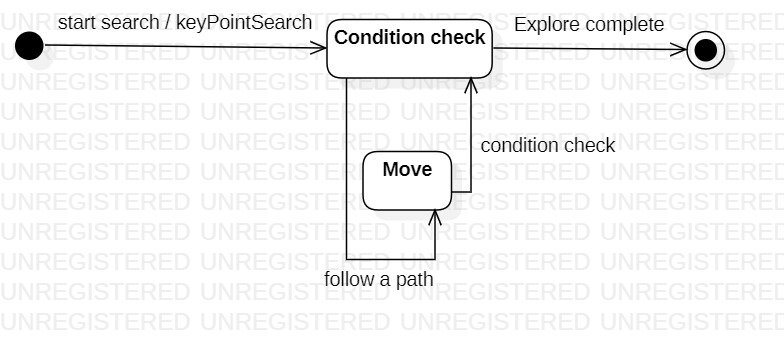
* 로봇을 전방으로 한 칸 이동시키는 기능을 한다.

**4 교류도(usecase : Start)**



keyPointStart 함수는 SimManager 내부에 있는 rotation, avoidHazard, detectingColorBlob, moveForward, compensationMotion 등의 명령을 끝날 때까지 수행해서 탐색 지점을 모두 탐색하는 기능을 한다. 맵 정보가 수정되어야 하는 경우 MapManager 클래스의 addHazard, addColorBlob을 불러 Map 클래스의 데이터를 수정시켜준다.

**5 SimManager 객체 상태도**



SimManager는 심에게 받아온 데이터를 이용해 주요지점을 방문하는 일을 하는 클래스이다. 다음 두개 중 하나의 상태를 가진다.

* Condition check
* 경로를 받아 이동하기 전에 현재 경로로 이동할 수 있는지 검사한다.
* 이동할 수 있는 경우 이동상태로 이동한다.
* 이동할 수 없는 경우 경로를 재설정하고 다시 조건을 검사한다.
* 만약 모든 탐색이 끝나면 종료한다.
* Move
* 해당하는 경로로 이동한다.
* 이동이 끝나면 다시 검사 상태로 돌아간다.