



다중 무인 체계 유무인 협동 시뮬레이터 개발

일시 2022년 11월 29일

Index.

다중 무인 체계 유무인
협동 시뮬레이터 개발

01

연구 개요

- 연구 목표
- 연구 일정 및 프로세스

02

연구 성과

- 목표 대비 성과

03

연구 내용

- S/W Framework Architecture
- 제어 S/W
- 명령 S/W
- 시뮬레이션 S/W
- 군집 제어 알고리즘 & 인공지능 모듈

01 연구 개요

01 연구 개요

연구 목표

AI-통합 시뮬레이터 구조 개발

개 요

과 제 명

다중 무인 체계 유무인 협동 시뮬레이터 개발

연구기간

'21.11 ~ '22.11 (12개월)

연구 목표

목 표

- 시뮬레이터 구조 모듈화
- S/W 내,외부 인터페이스 설계
- AI 학습 모듈 분리
- 군집 임무 제어 시나리오를 통한 시뮬레이터 검증

연구 성과물

성 과 물

- 모듈화된 시뮬레이터 S/W
- ICD 정의 기반 S/W 내,외부 인터페이스
- 군집 임무 제어 알고리즘



01 연구 개요

• 연구 일정

연구 개발 항목	추진 일정 (개월)											
	12	1	2	3	4	5	6	7	8	9	10	11
▪ 단일 객체 모델 개발 ✓ S/W 구조 모듈화 ✓ S/W 데이터 송·수신을 위한 I/O 인터페이스 구축												
▪ 복수 객체 모델 개발 ✓ 제어 S/W 내부 확장성과 변경 가능성을 고려한 S/W 아키텍처 설계 ✓ 복수 객체 모델을 고려한 통신 인터페이스 확장 구축 ✓ 복수 객체 모델에 대한 명령,시뮬레이션 S/W 설계 및 구현												
▪ AI 모듈 설계 및 군집 임무 계획 연구 ✓ 시뮬레이션 S/W의 구조 확장 ✓ AI 학습이 가능한 모듈의 상세 설계												
▪ 시나리오에 따른 시뮬레이터 검증 및 평가												

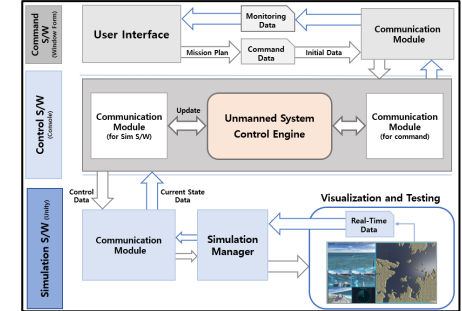
02 연구 성과

02 연구 성과

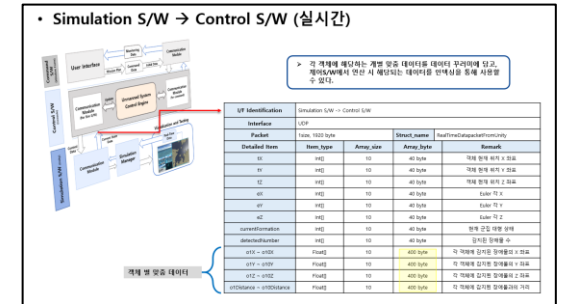
• 목표 대비 성과

수행 항목	목표 대비 성과
시뮬레이터 구조 모듈화	<ul style="list-style-type: none"> 명령 S/W, 제어 S/W, 시뮬레이션 S/W 총 3개 부분으로 구조화 시뮬레이션 엔진에 해당하는 제어 S/W의 SOLID 원칙 기반 설계로 새로운 객체, 환경 및 알고리즘의 추가, 변경, 제거가 용이
S/W 내·외부 인터페이스 설계	<ul style="list-style-type: none"> 모듈화 된 S/W 사이의 통신 인터페이스, S/W 내부 모듈 간의 인터페이스 설계 및 구현 완료
AI 학습 모듈 설계	<ul style="list-style-type: none"> .onnx (Open Neural Network Exchange) 파일을 활용한 해검의 객체 인식 알고리즘 및 강화학습 기반 지역 탐색 알고리즘 구현
군집 임무계획 알고리즘을 이용한 시뮬레이터 검증	<ul style="list-style-type: none"> 모듈화 된 S/W에서 제어 S/W에 Leader-Follower 기반 군집 임무계획 알고리즘 구현 간단한 시나리오를 통한 검증 완료

◆ 제어 S/W를 중심으로 한 모듈화된 소프트웨어 구조



◆ S/W간 내/외부 인터페이스 설계



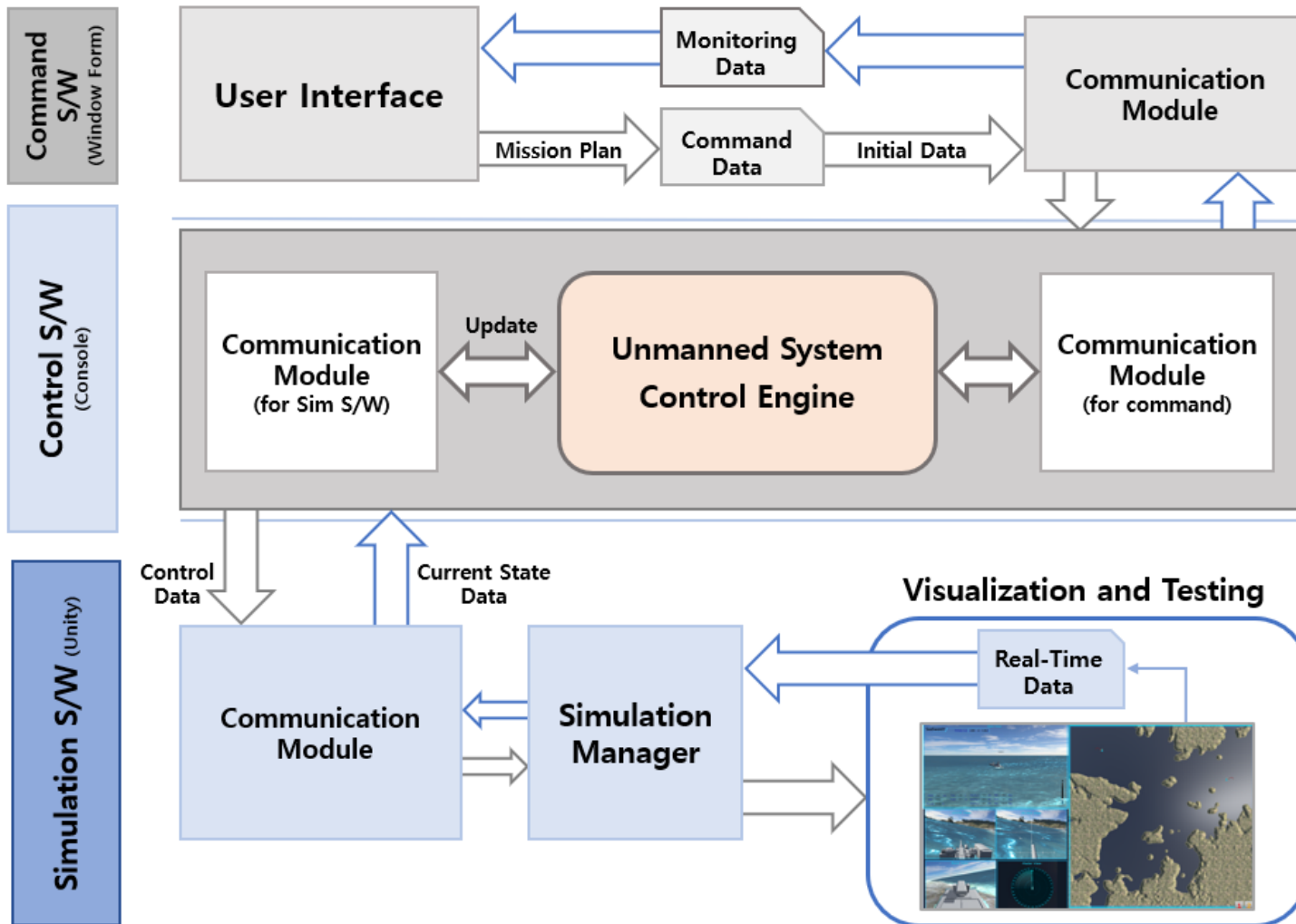
◆ 군집 제어 알고리즘 구현



03 연구 내용

03 연구 내용

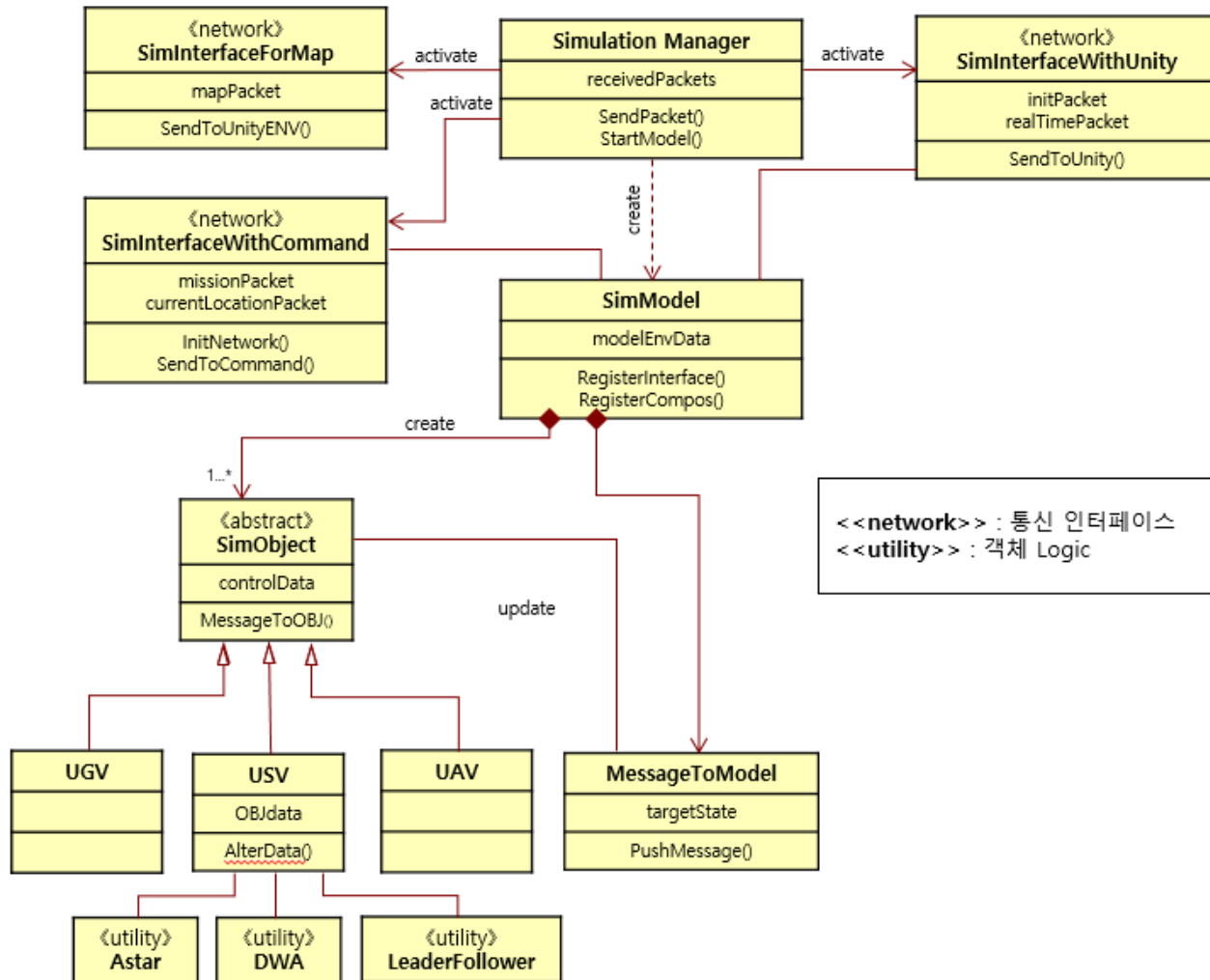
• 시뮬레이션 S/W Framework 구조도



- ✓ 제어 S/W를 중심으로 하는 총 3개의 **모듈화 된 S/W**의 조합으로 이루어진 시뮬레이터 구성
- ✓ 각 S/W에 있는 **통신 모듈을 통해** 모든 데이터 취합 및 송수신
- ✓ **명령 S/W** 에서 UI를 통해 시뮬레이션 시나리오 입력
- ✓ **제어 S/W**에서 내부 로직 및 알고리즘을 사용하여 무인 체계에 대한 제어 담당
- ✓ **시뮬레이션 S/W**에서는 제어 S/W에서 받은 제어 입력 데이터를 기반으로 시뮬레이션 수행 및 가시화, 실시간 데이터는 제어 S/W로 다시 송신

03 연구 내용

• 제어 S/W 구조 - 클래스 다이어그램



- ✓ 제어 S/W 간략화 된 클래스 다이어그램
- ✓ Simulation Manager 에 있는 메인 메소드로 **프로그램 시작**, Simulation Manager는 시뮬레이션의 핵심이 되는 SimModel과 통신에 필요한 `<<network>>` 클래스들을 **생성 및 활성화**
- ✓ SimModel 은 시뮬레이션 대상 객체들의 **집합**으로, 객체들을 **생성**하고, 통신 인터페이스를 통해 수신 받은 **데이터 분배** 및 제어 입력 **데이터 취합**
- ✓ SimObject를 추상 클래스로 상속받는 USV, UAV, UGV 같은 클래스들은 시뮬레이션 시작 시 생성되며, 각자 가지고 있는 내부 Logic에 따라 **제어 입력 데이터 도출**
- ✓ 객체들에 대한 Logic 변경 시 USV,UAV같은 클래스들의 내부 알고리즘 파트만 수정한다면 **Logic 변경 가능**
- ✓ 새로운 플랫폼에 대한 시뮬레이션 수행 시, **새로운 클래스 정의 후 구현한다면** SimModel에서 객체 생성 후 시뮬레이션 가능
- ✓ Solid 원칙 기반 내부 설계

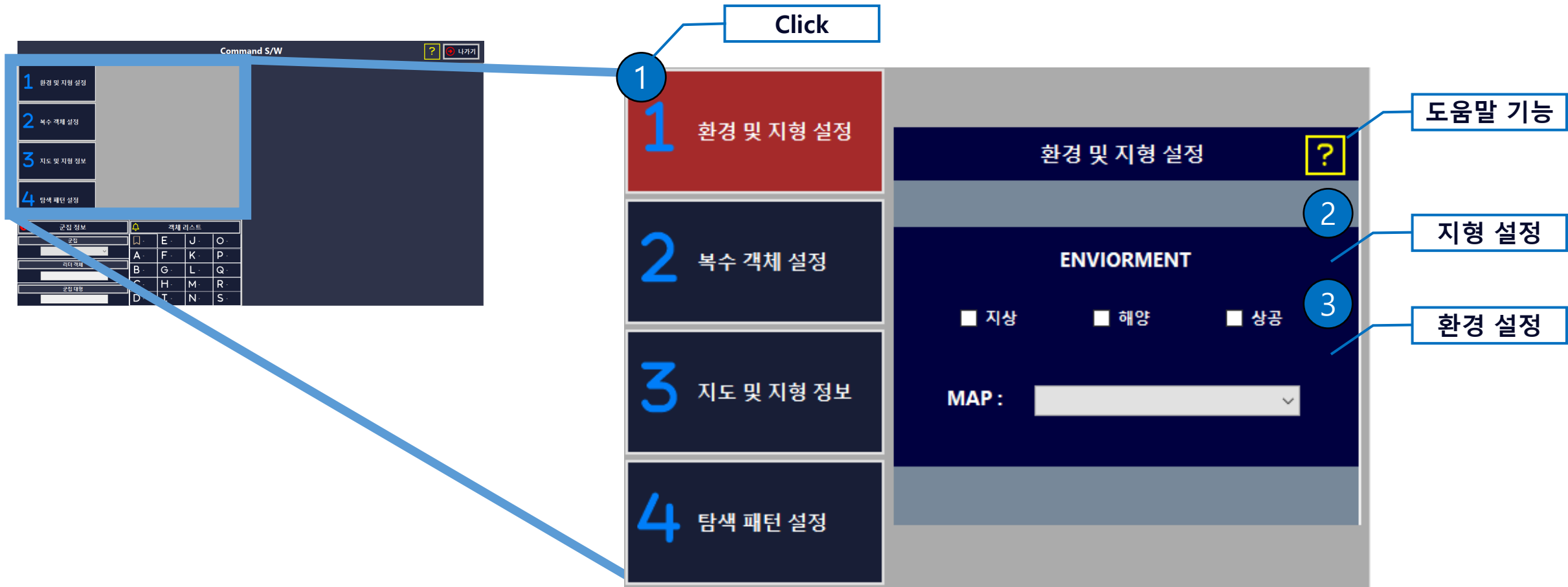
03 연구 내용

• 명령 S/W 레이아웃 및 기능



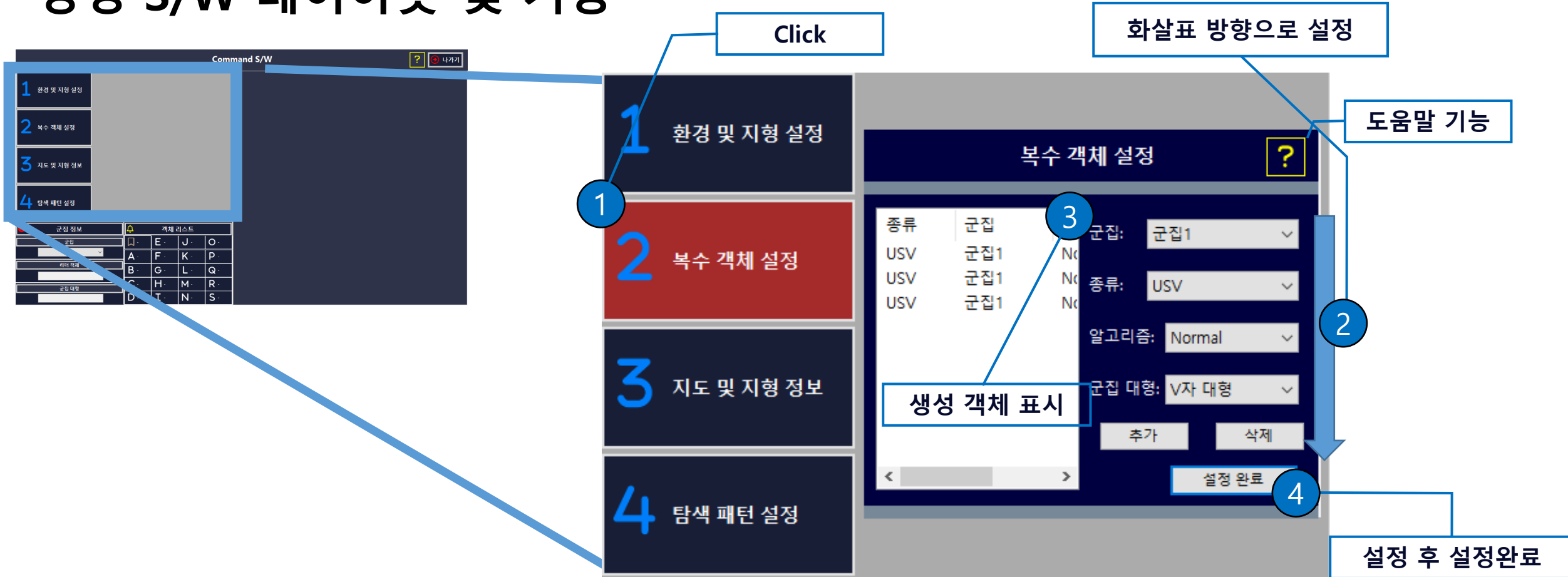
03 연구 내용

• 명령 S/W 레이아웃 및 기능



03 연구 내용

• 명령 S/W 레이아웃 및 기능



✓ 객체 설정 후 Form간 데이터 전달 가능

✓ 객체 마다 설정하고 싶은 속성 자유롭게 추가 및 수정 가능

03 연구 내용

• 명령 S/W 레이아웃 및 기능

Command S/W

초기 위치 설정

위치정보 할당

실시간 위치 확인

Click

객체 범례 표시

객체 위치 설정

1 환경 및 지형 설정

2 복수 객체 설정

3 지도 및 지형 정보

4 탐색 패턴 설정

지도 및 지형 정보

5 통신 및 시뮬레이션 시작

6 실시간 정보 불러오기

7 객체 선택

군집 정보

군집

리더 객체

군집 대형

객체 리스트

USV1	E -	J -	O -
A USV2	F -	K -	P -
B USV3	G -	L -	Q -
C -	H -	M -	R -
D -	I -	N -	S -

출발 위치

도착 위치

리더 중심 전역경로 생성

USV2

나가기

14

03 연구 내용

• 명령 S/W 레이아웃 및 기능

The diagram illustrates the layout and functionality of the Command S/W interface. It is divided into two main sections: a main menu on the left and a detailed settings window on the right.

Main Menu Layout:

- 1 환경 및 지형 설정** (Environment and Terrain Setting)
- 2 복수 객체 설정** (Multiple Object Setting)
- 3 지도 및 지형 정보** (Map and Terrain Information)
- 4 탐색 패턴 설정** (Search Pattern Setting)

Search Pattern Setting Window:

- 탐색 패턴 설정** (Search Pattern Setting) - Title bar with a help icon.
- 객체 선택** (Object Selection) - A list of objects with checkboxes: ☒ USV1, ☒ USV2, ☐ USV3.
- 탐색 패턴 선택** (Search Pattern Selection) - A dropdown menu showing options: Grid, Grid, Linear, Quadrant or Zone, Spiral. A mouse cursor is shown clicking on the first 'Grid' option.
- 설정 완료** (Setting Complete) - A button at the bottom right.

Annotations:

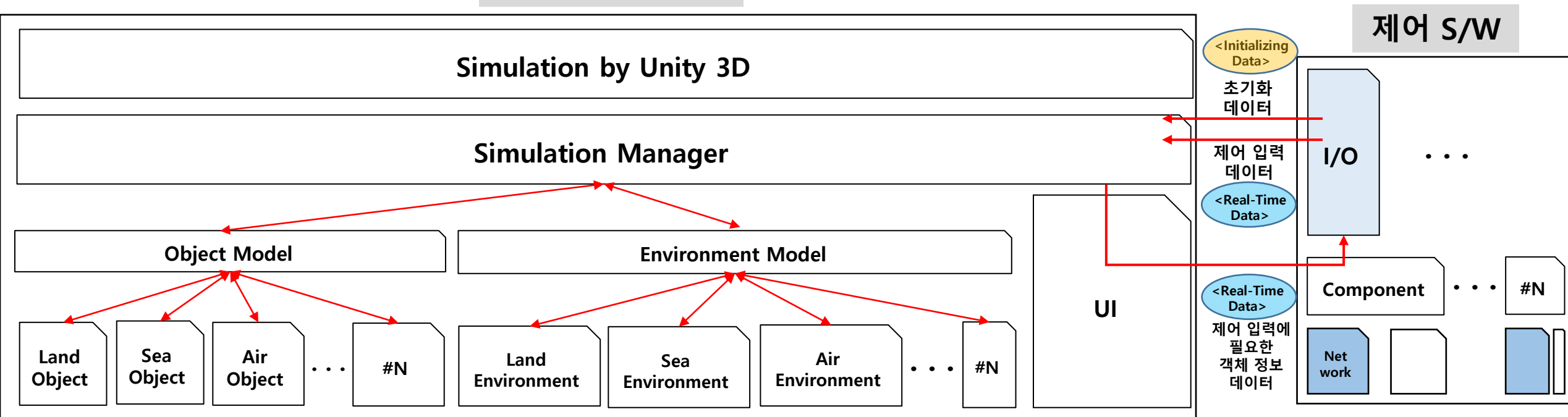
- A blue box on the left contains a list of menu items: 1 환경 및 지형 설정, 2 복수 객체 설정, 3 지도 및 지형 정보, 4 탐색 패턴 설정. A blue arrow points from this box to the main menu.
- A blue circle with the number '1' is placed over the '탐색 패턴 설정' menu item.
- A blue circle with the number '2' is placed over the 'USV2' checkbox in the object selection list.
- A blue circle with the number '3' is placed over the first 'Grid' option in the search pattern dropdown.
- A blue circle with the number '4' is placed over the '설정 완료' button.
- A blue box labeled 'Click' has an arrow pointing to the '탐색 패턴 설정' menu item.
- A blue box labeled '탐색 패턴 선택' has an arrow pointing to the search pattern dropdown.
- A blue box labeled '설정 후 설정완료' has an arrow pointing to the '설정 완료' button.

- ✓ 필요에 따라 목표 지점 도달 후 탐색 패턴 설정 가능
- ✓ 객체마다 설정하고 싶은 탐색 패턴 자유롭게 추가 및 수정 가능

03 연구 내용

• 시뮬레이션 S/W 구조 (Unity 3D)

시뮬레이션 S/W



- Unity 내부의 Simulation Manger는 시뮬레이션 S/W를 제어할 수 있음
- Simulation Manger는 제어 S/W로 부터 초기화 데이터 수신 후, Object + Enviornment Model 을 활용해 **다양한 환경의 객체 및 지형**을 동적 생성. 이후 각 Component들은 객체 상황 데이터를 Simulation Manger로 보냄
- Simulation Manger가 데이터를 취합하여 제어 S/W로 객체 상황 데이터를 보내고, 제어 입력 데이터를 수신함.

03 연구 내용

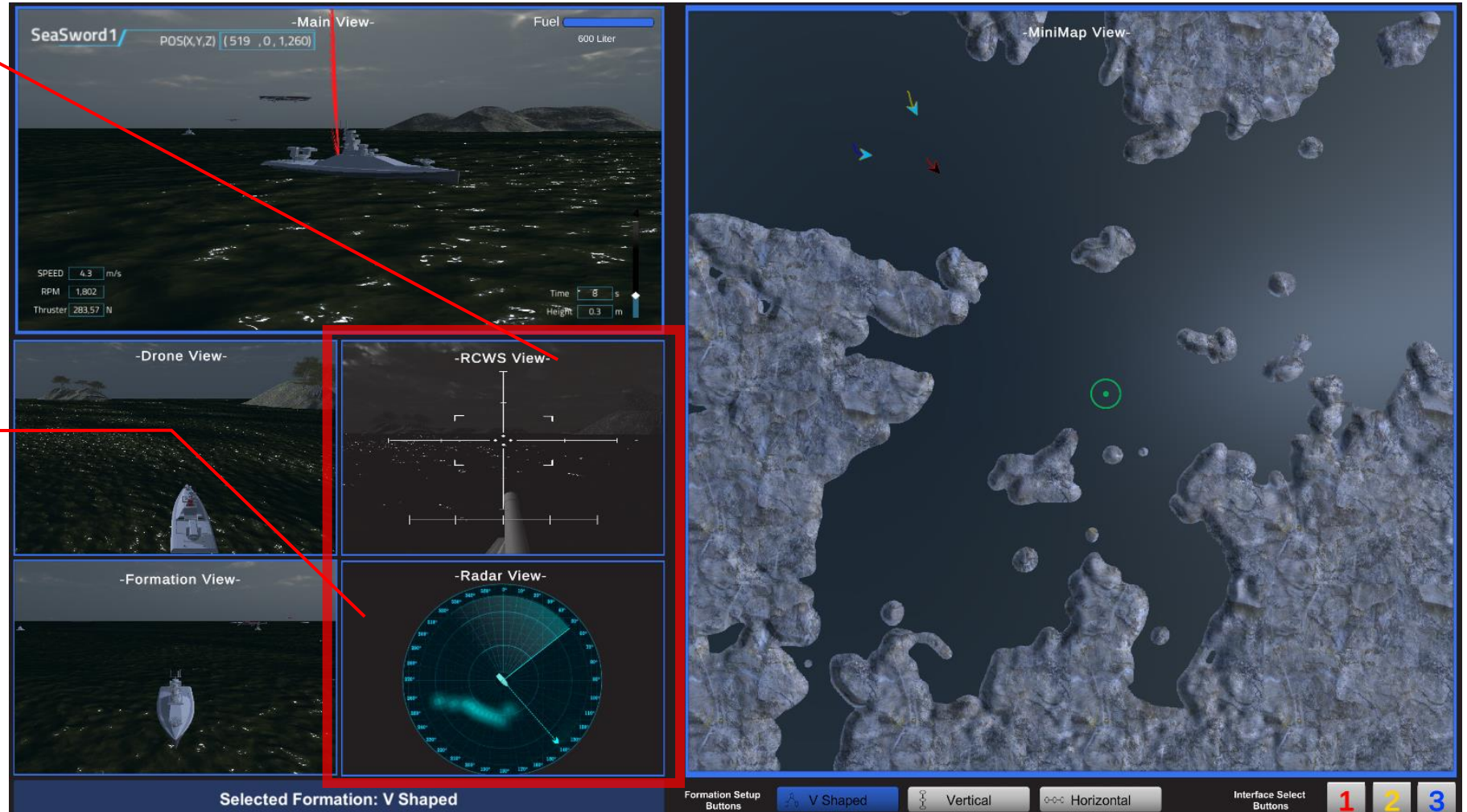
• 시뮬레이션 S/W UI 화면 예시 – 인터페이스 1

RCWS View

RCWS 화면을
EO/IR 카메라로
확인할 수 있음

Radar View

리더 객체에 레이더가
구현되어,레이더에서 검출
되는 물체를 객체 기준으로
확인할 수 있음



03 연구 내용

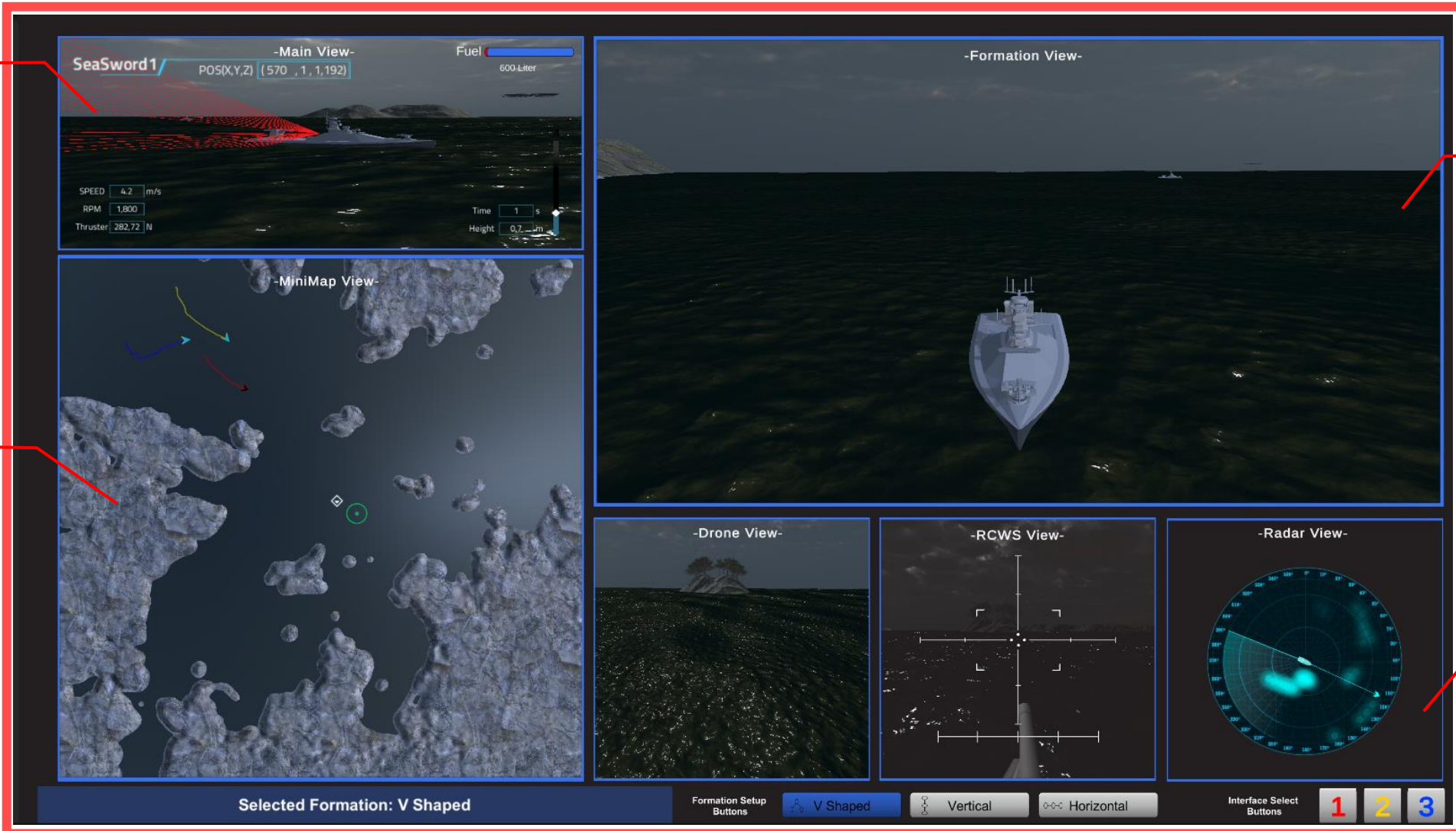
• 시뮬레이션 S/W UI 화면 예시 – 인터페이스 2

Main View

↓
객체를 확인하기
위해 필요한
정보들을 담은
메인 화면

MiniMap View

↓
각 객체들의
이동경로 및
도착 지점을
확인할 수 있음



Formation View

↓
리더를 기준으로
주변 객체들의
대형을 한 눈에
확인할 수 있음

Drone,
RCWS,
Radar View

03 연구 내용

• 시뮬레이션 S/W UI 화면 예시 – 인터페이스 3



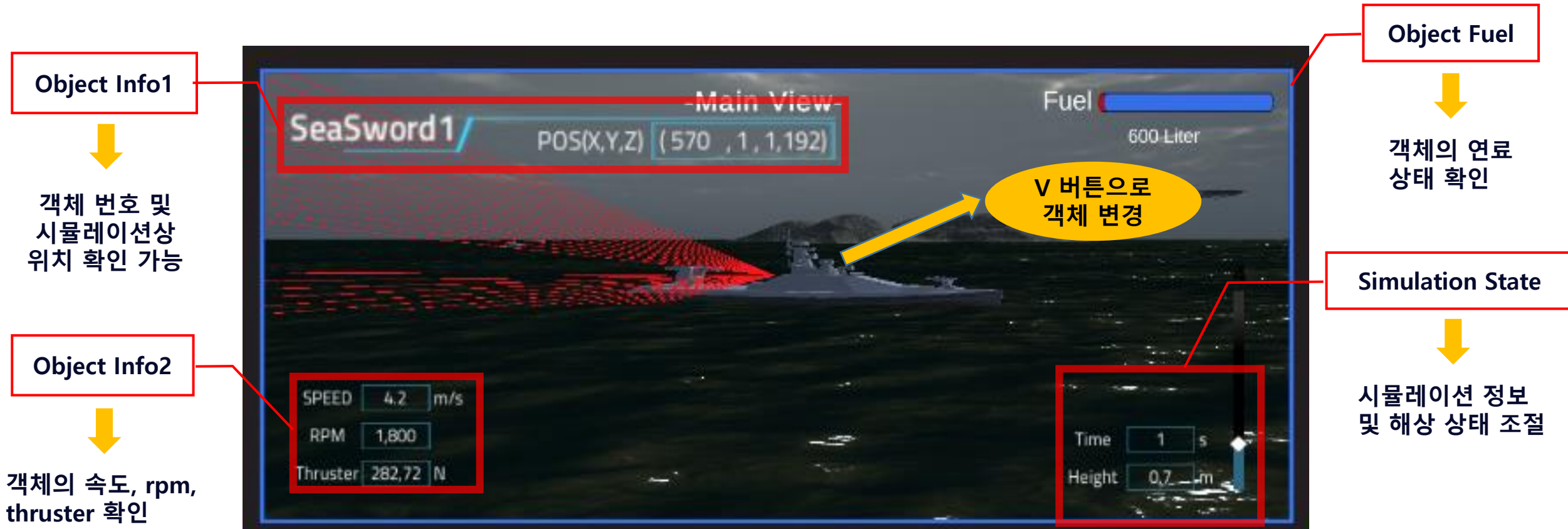
Drone View



각 객체의 드론을 기준으로
지형지물과 대형을 확인할
수 있음
→ 해상과 공중이 상호작용
하는 모습 확인 가능

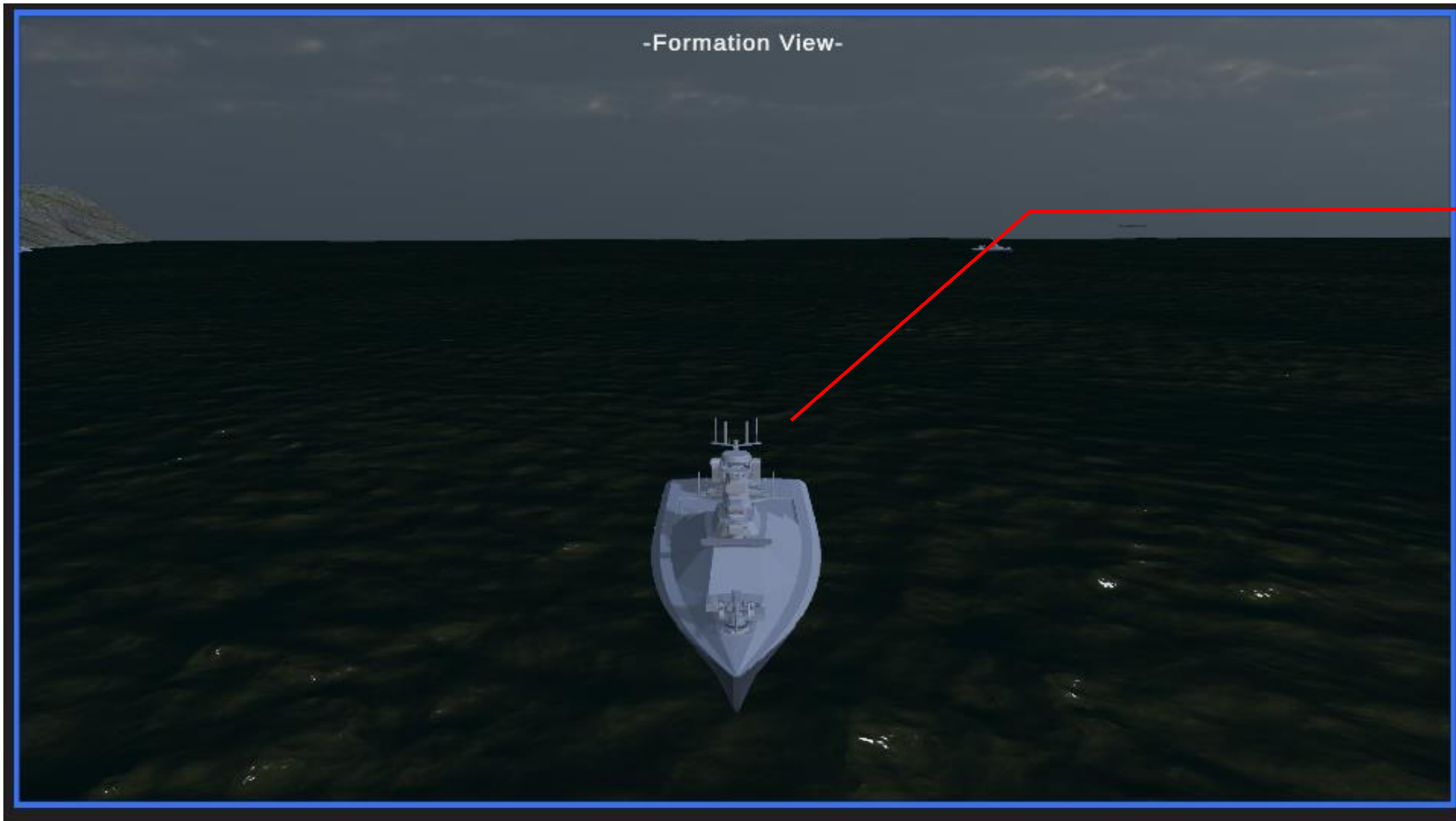
03 연구 내용

• 시뮬레이션 S/W UI 화면 예시 – Main View



03 연구 내용

- 시뮬레이션 S/W UI 화면 예시 – Formation View



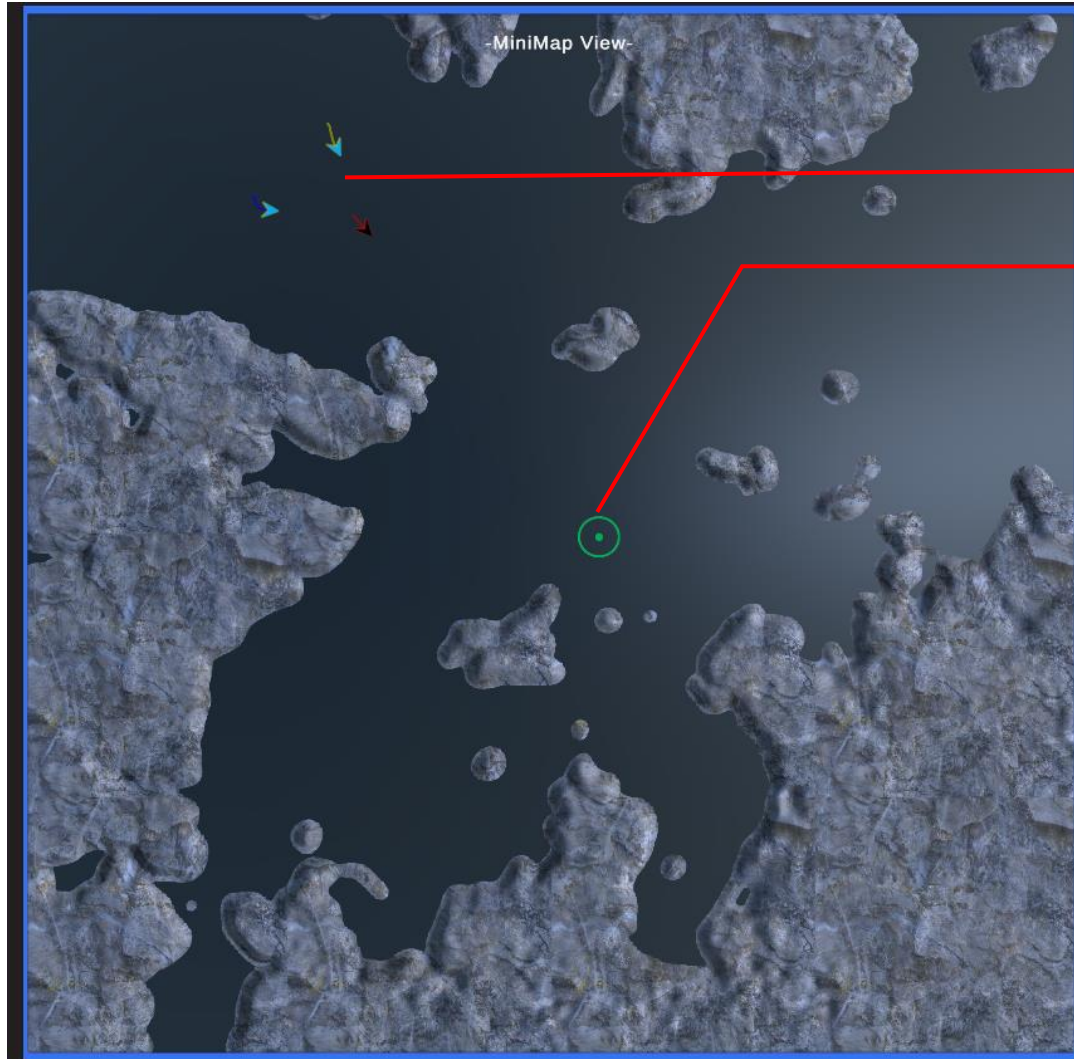
Formation View



리더 객체를 기준으로 주변 군집
대형이 어떻게 되는지 화면
상에서 확인할 수 있음

03 연구 내용

• 시뮬레이션 S/W UI 화면 예시 – MiniMap View



Mission Destination



명령 S/W에서 군집 객체들의
목적지를 정하면, 시뮬레이션
S/W UI의 MiniMapView에
다음과 같은 아이콘으로
목적지가 표시됨

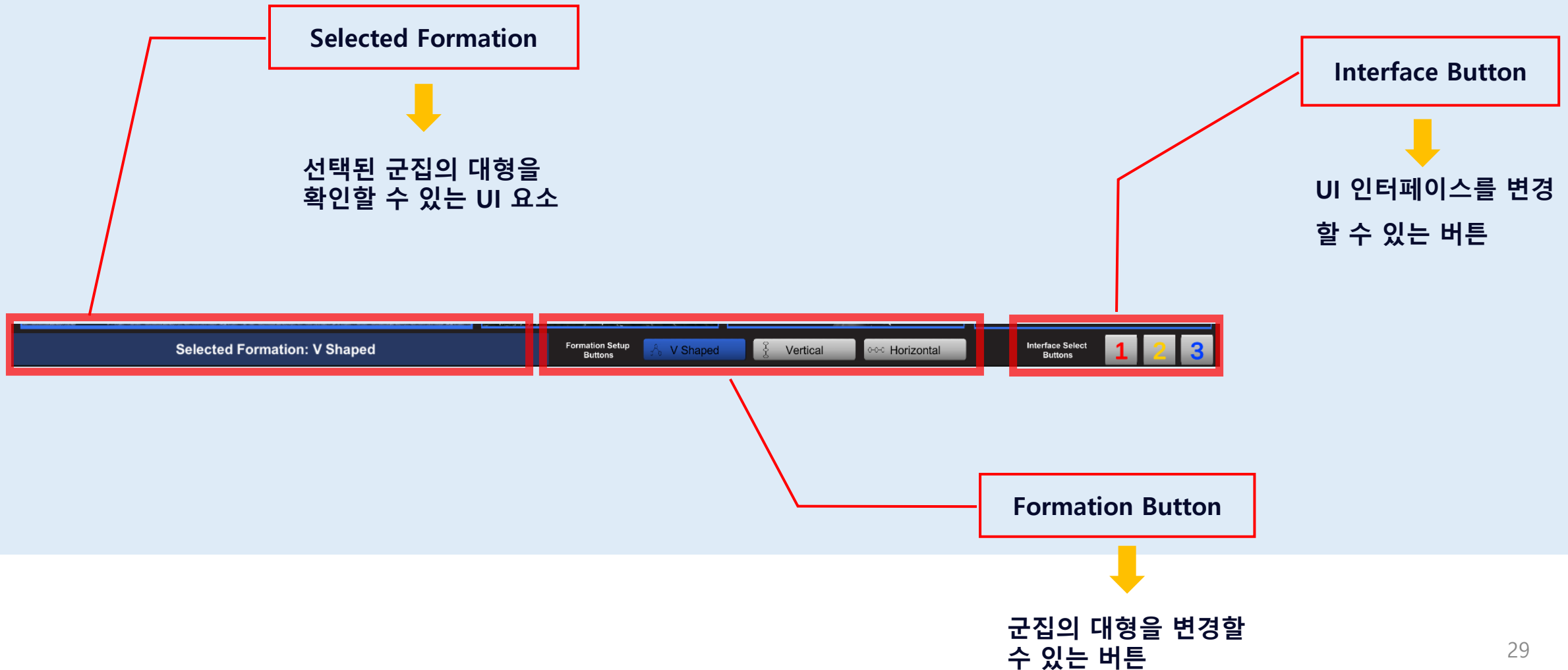
Object Route



각 객체들이 MiniMap
화면에서 어디에 위치되어
있는지 확인이 가능하고,
각 객체들의 이동경로를
각기 다른 색들의 이동 경로로
확인할 수 있음

03 연구 내용

• 시뮬레이션 S/W UI 화면 예시 - Button



03 연구 내용

- **시뮬레이션 S/W Environment**

수중 환경

초기 화면

Scene Loading...

지상

공중

해상

- 초기 화면을 시작으로 명령 S/W의 시나리오 입력에 따라 Scene 단위로 구성되어있는 다양한 환경을 불러올 수 있음

- ✓ 동역학적 모델링 어려움으로 인해
수중 환경 구축 제한
- ✓ 추후 환경 구축 시 시뮬레이션 S/W 내부에
Scene 추가를 통해 손쉽게 환경 추가 가능

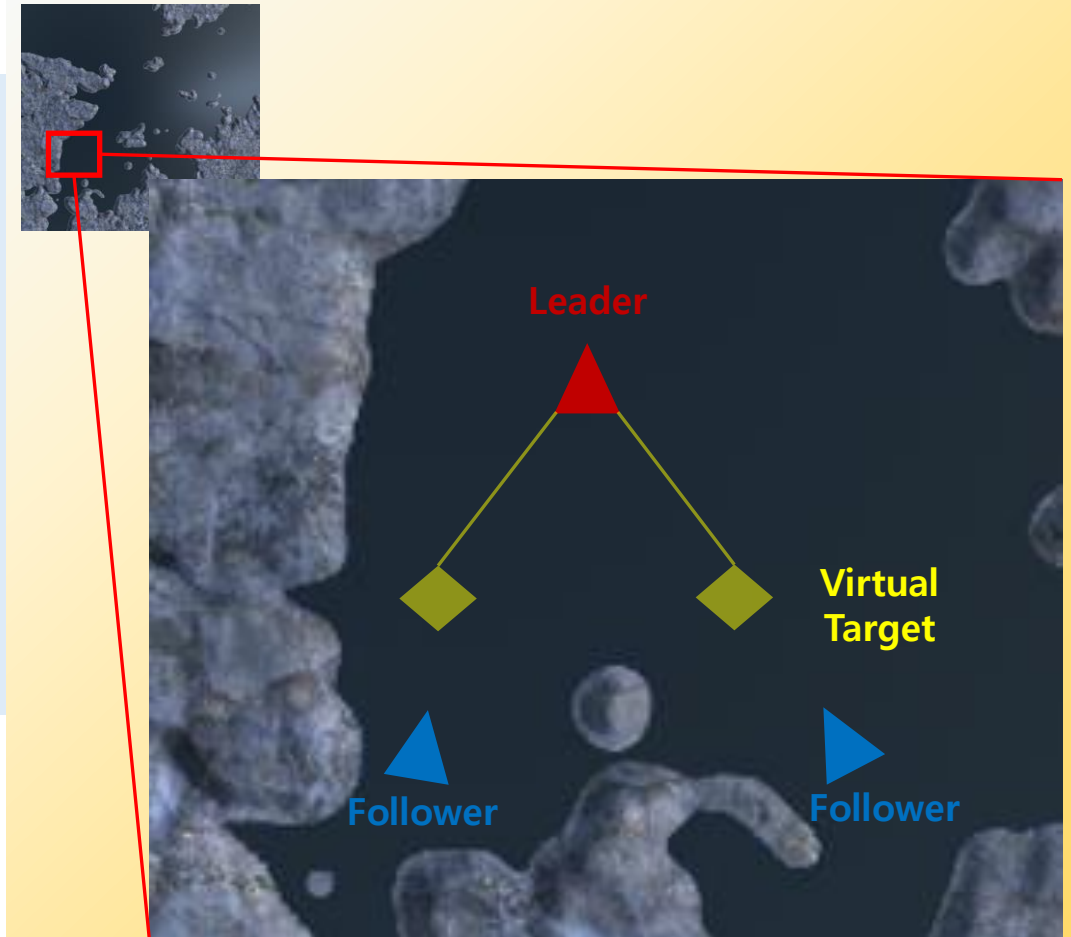
2C6H6 추가를 통해 순전히 화석 추가 가능
 ^ 순환 화석 수준 Y Y리튬다이옥 2M 대금에

03 연구 내용

• 군집 제어 알고리즘 – Leader-Follower Algorithm

- ✓ 시뮬레이터의 군집 제어 임무 모의 기능을 검증하기 위해 **Leader – Follower Algorithm** 구현
- ✓ 임무에서의 전역 경로를 따라 가는 Leader의 상태 정보를 기반으로 설정한 대형을 유지
- ✓ Follower 는 Leader 의 상태 정보를 기반으로 생성된 각자의 **가상의 타겟**을 목적지로 설정하여 이동

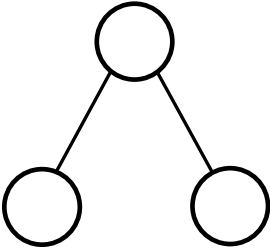

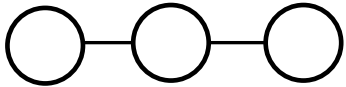
➤ Leader – Follower with Virtual Target



03 연구 내용

• 군집 제어 알고리즘 구현 – Leader-Follower Algorithm

➤ 가상 Target 생성 Equation

1. V-Shaped	<div>If robot's number is odd$X_f = X_l + (Int(robotnum/2) + 1) \times Distance \times Cos(\theta)$$Y_f = Y_l + (Int(robotnum/2) + 1) \times Distance \times Sin(\theta)$$\theta = \varphi_l + \Pi - V_{shapeAngle}$</div> <div>If robot's number is even$X_f = X_l + Int(robotnum/2) \times Distance \times Sin(\theta)$$Y_f = Y_l + Int(robotnum/2) \times Distance \times Cos(\theta)$$\theta = \varphi_l + \Pi + V_{ShapeAngle}$</div>	
2. Vertical	$X_f = X_l + (robotnum) \times distance \times 4/3 \times Sin\theta$ $Y_f = Y_l + (robotnum) \times distance \times 4/3 \times Cos\theta$ $\theta = \varphi_l + \Pi$	
3. Horizontal	<div>If robot's number is odd$X_f = X_l + (Int(robotnum/2) + 1) * Distance * Sin\theta$$Y_f = Y_l + (Int(robotnum/2) + 1) * Distance * Cos\theta$$\theta = \varphi_l + \Pi + \Pi/2$</div> <div>If robot's number is even$X_f = X_l + Int(robotnum/2) * Distance * Sin\theta$$Y_f = Y_l + Int(robotnum/2) * Distance * Cos\theta$$\theta = \varphi_l + \Pi - \Pi/2$</div>	

03 연구 내용

• 군집제어 알고리즘 구현- Leader-Follower Algorithm

➤ Sensitivity Reduction

- ✓ Follower의 가상 타겟을 설정하였는데, Leader의 방향각이 바뀔 때, **거리가 멀리 있는 Follower는 가상 Target의 위치 변화가 매우 큰 문제점이 발생**
- ✓ 이에 Leader의 방향각(Euler 각)에 따른 Follower의 가상 Target 변동을 줄이기 위해 **Sensitivity Reduction 및 Smoothing** 적용
- ✓ Simple Moving Average(3) 사용
- ✓ 또한 가상 Target 생성시 **리더의 진행 방향을 고려하기 위해서** 가상 Target 생성 Equation에 Leader의 현재 위치를 함께 넣어 진행 방향도 참고할 수 있게 Target Position 생성

➤ Simple Moving Average

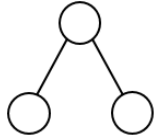


$$SMA = \frac{A_1 + A_2 + \dots + A_n}{n}$$

Simple Moving Average(3)

$$T_t = (T_{t-1} + T_{t-2} + T_{t-3}) \div 3$$

T_t = Virtual Target at Time t

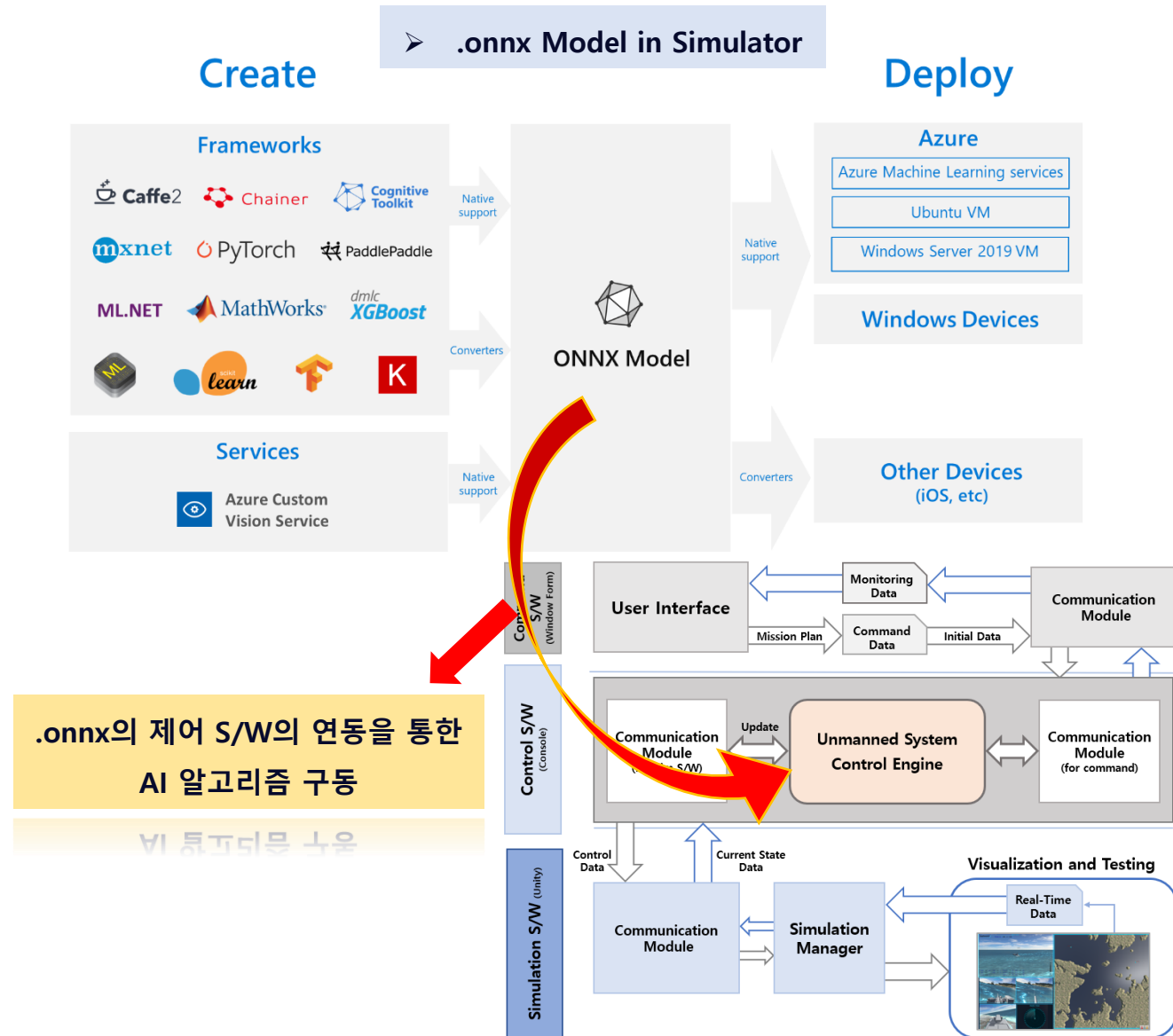
➤ Equation of Generating Virtual Target

Final Virtual Target		
1. V-Shaped $X_f = 0.8X_f + 0.2X_l$ $Y_f = 0.8Y_f + 0.2Y_l$ X_f = Follower's X coordinate of Target X_l = Leader's X coordinate of Target		2. Vertical $X_f = 0.8X_f + 0.2X_l$ $Y_f = 0.8Y_f + 0.2Y_l$ X_f = Follower's X coordinate of Target X_l = Leader's X coordinate of Target
		3. Horizontal $X_f = 0.6X_f + 0.4X_l$ $Y_f = 0.6Y_f + 0.4Y_l$ X_f = Follower's X coordinate of Target X_l = Leader's X coordinate of Target
		

03 연구 내용

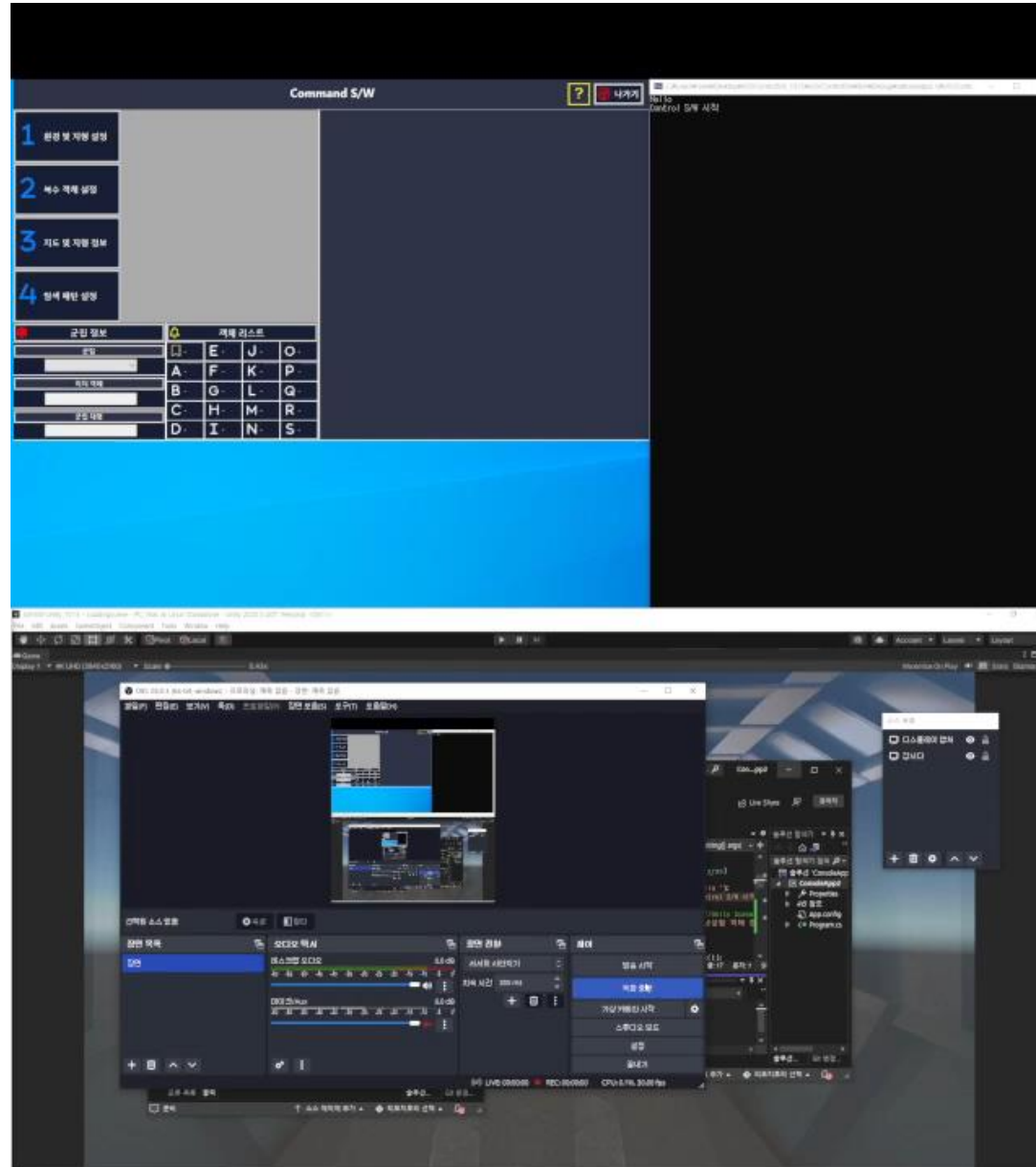
• AI 모듈

- ✓ 본 시뮬레이터에는 AI 알고리즘에 대한 학습 및 검증 기능이 요구되었기 때문에 AI 알고리즘 파일을 제어 S/W에서 연동하여 구동할 수 있어야 함
- ✓ C# Nuget 패키지 매니저에 있는 Microsoft.ML 라이브러리가 범용 Neural Net 확장자인 .onnx(Open Neural Network Exchange) 파일에 대한 서비스를 지원하기 때문에 이를 활용하여 AI 알고리즘을 제어 S/W에 구현하고자 함
- ✓ 본 연구에서는 .onnx 확장자에 맞춰 해검의 객체 인식 알고리즘(YOLO - Based), 강화학습 기반 지역 탐색 알고리즘을 구현하였음



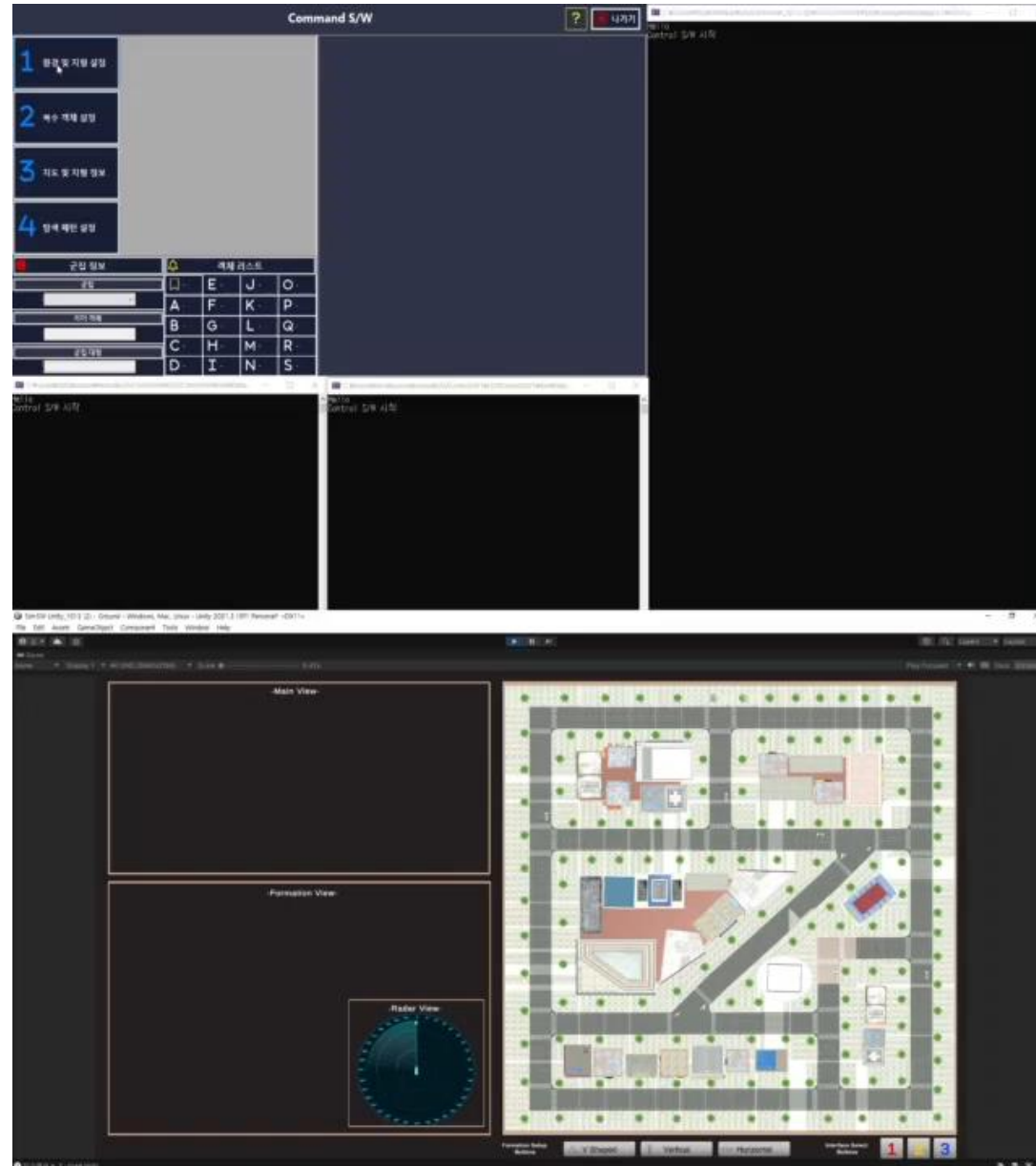
03 연구 내용

- 연구 결과물 영상



03 연구 내용

- 연구 결과물 영상





감 사 합 니 다