

# Missile Simulator

## MATLAB Programming Term Project

Team 2 : 백창인 권우석 김선용 박준우 허선

# Contents

- Intro
- Project Overview
- Development Process & Code-review
- Demonstration
- Q&A

# Intro

## <백창인>

- Project Management
- main.m, getTrack.m, isHit.m
- Class diagram & Flow chart
- getTrack subsystem development
- missile head direction development
- PPT & Presentation

## <권우석>

- getMissilePos.m
- getNextMissilePos.m
- Investigate Matlab to VR Coordinates
- Advice project report

## <김선용>

- getMissileDirection.m
- Investigate 3D Visualization
- Advice project report
- Report Assignment preparation

## <박준우>

- getFighterPos.m
- Investigate Matlab to VR Coordinates
- Write & Edit project Report

## <허선>

- displayPlot.m
- Investigate 3D Visualization
- 3D world & Visualization development
- Matlab to VR Coordinates development
- Explosion effect development

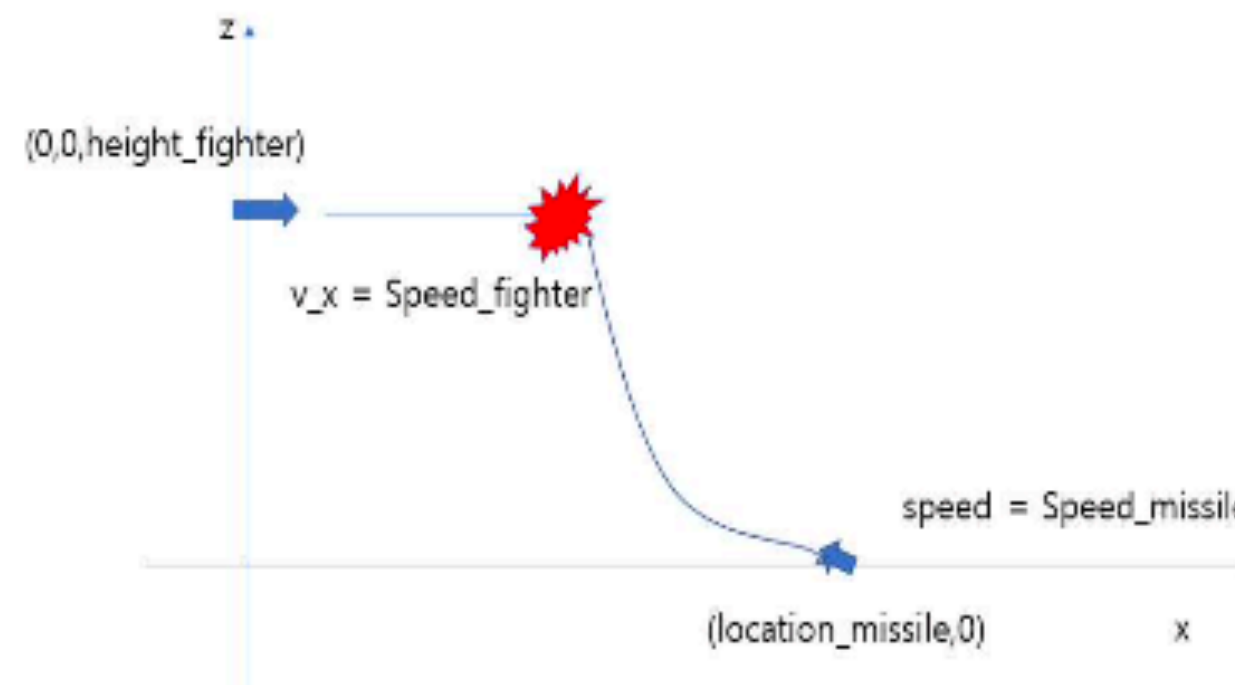
# Project Overview

## 2. 문제

다음과 그림과 같이 x와 z 축으로 구성된 평면에서 시뮬레이션 하려고 한다.

(0, height\_fighter)에 위치한 전투기가 속도 Speed\_fighter의 속도로 +x 방향으로 정속도 이동을 하고, 출발과 동시에 (location\_missile, 0)의 위치에 있는 미사일이 발사되어 미사일을 향해 속도 Speed\_missile으로 비행한다. 순간순간의 미사일의 방향은 그 순간의 전투기의 위치와 미사일의 위치를 연결하는 벡터 방향으로 진행한다. 미사일의 위치가 전투기의 위치좌표에서 5미터 안으로 들어오면 미사일이 명중한 것으로 간주하고 프로그램이 종료되어야 한다. 높이와 위치의 단위는 km이고, 속도는 km/sec로 받는다.

(대표값 전투기위치(0, 0, 100), 속도 30km/sec, 미사일위치 (500, 0, 0), 속도 60)



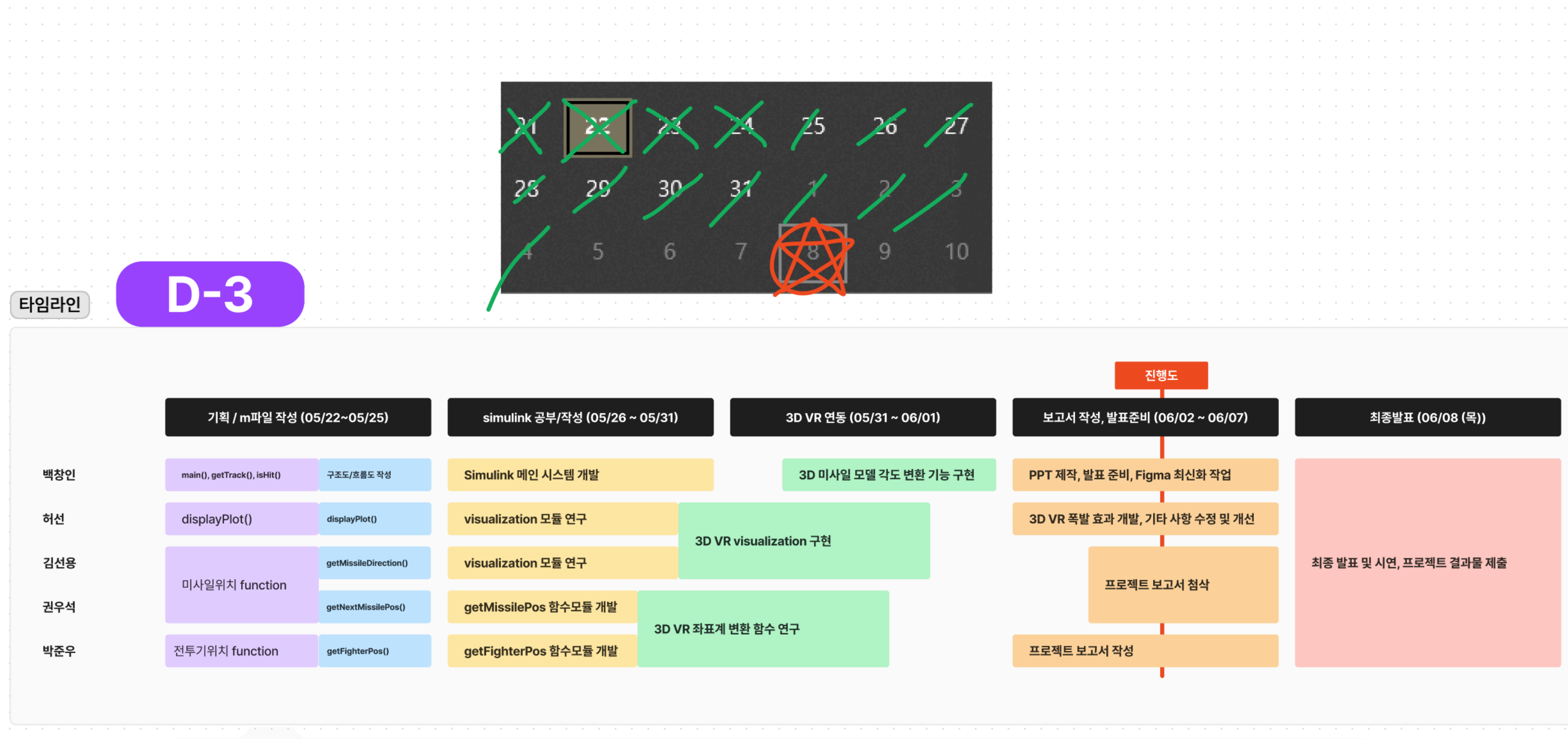
전투기의 초기 위치, 속도, 방향  
미사일의 초기 위치, 속도, 방향  
을 입력받아,

미사일과 전투기의 움직임을  
명중 시까지 Simulation 하여  
3D VR Visualization 이 최종 목표

최종 목표는 3D VR로 전투기와 미사일이 움직이는 모양을 애니메이션을 통해 보는 것이며, 격추시에 격추 표시는 option임.

# Development Process

## Figma

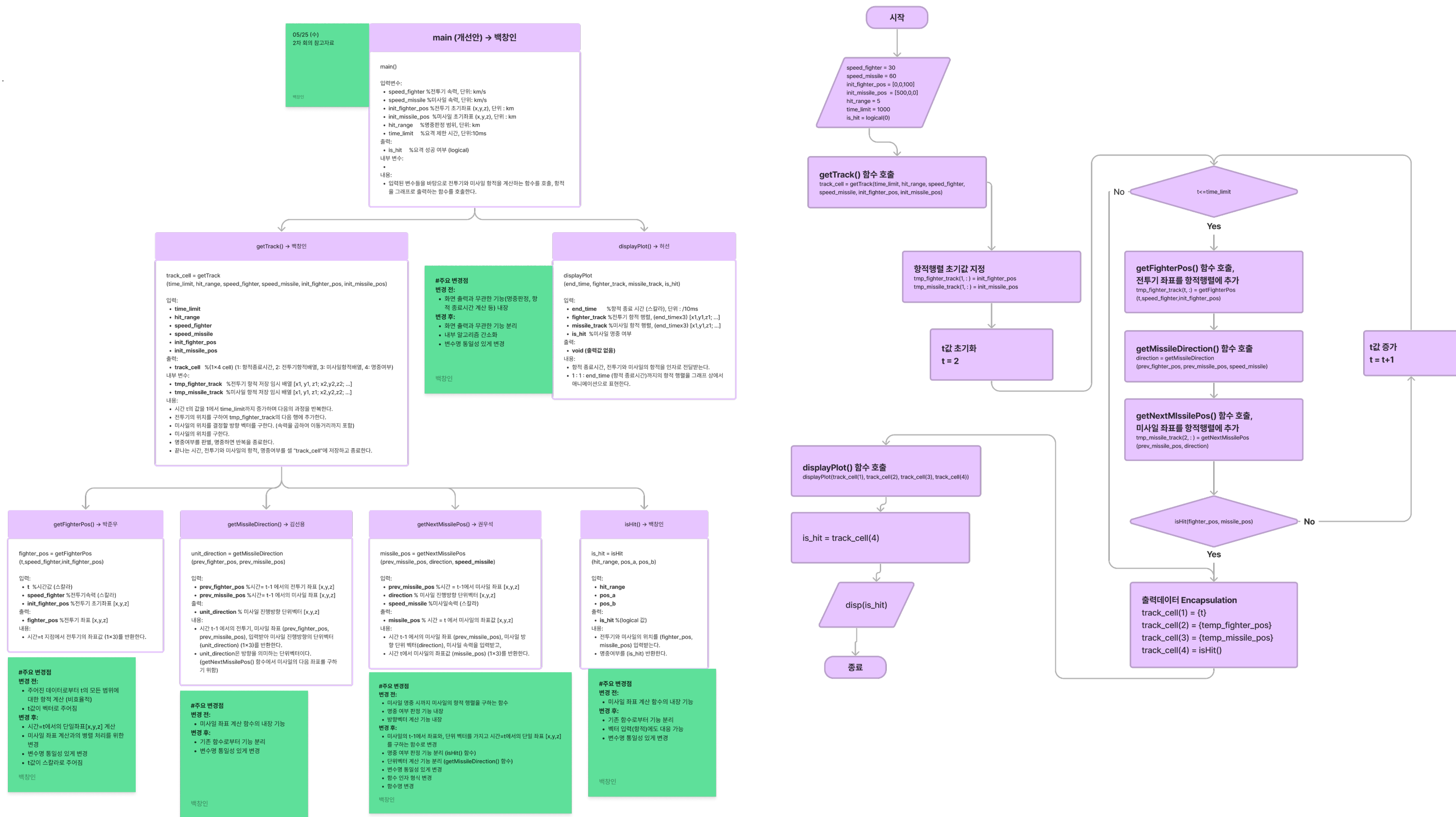


- 웹 기반 협업 tool, 함께 사용할 수 있는 Figjam 활용
- 프로젝트 일정관리, 시각화 & 프로그램 구조도 작성 등

# Development Process

## 기획 단계

<구조도 & 흐름도>



- Class Diagram, Flow chart 제작, 개발 일정 수립, 역할 분담



# Development Process

## M-file

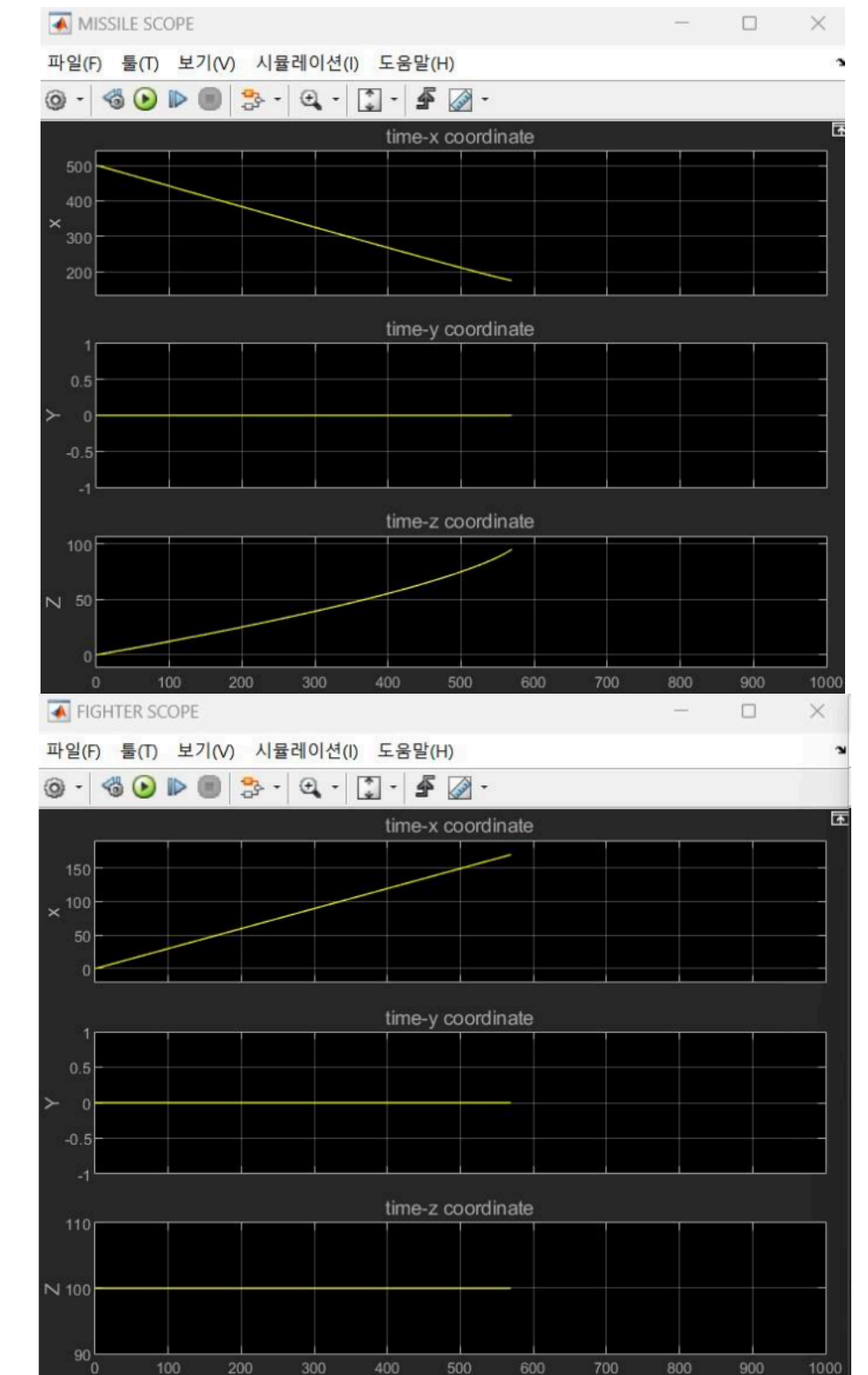
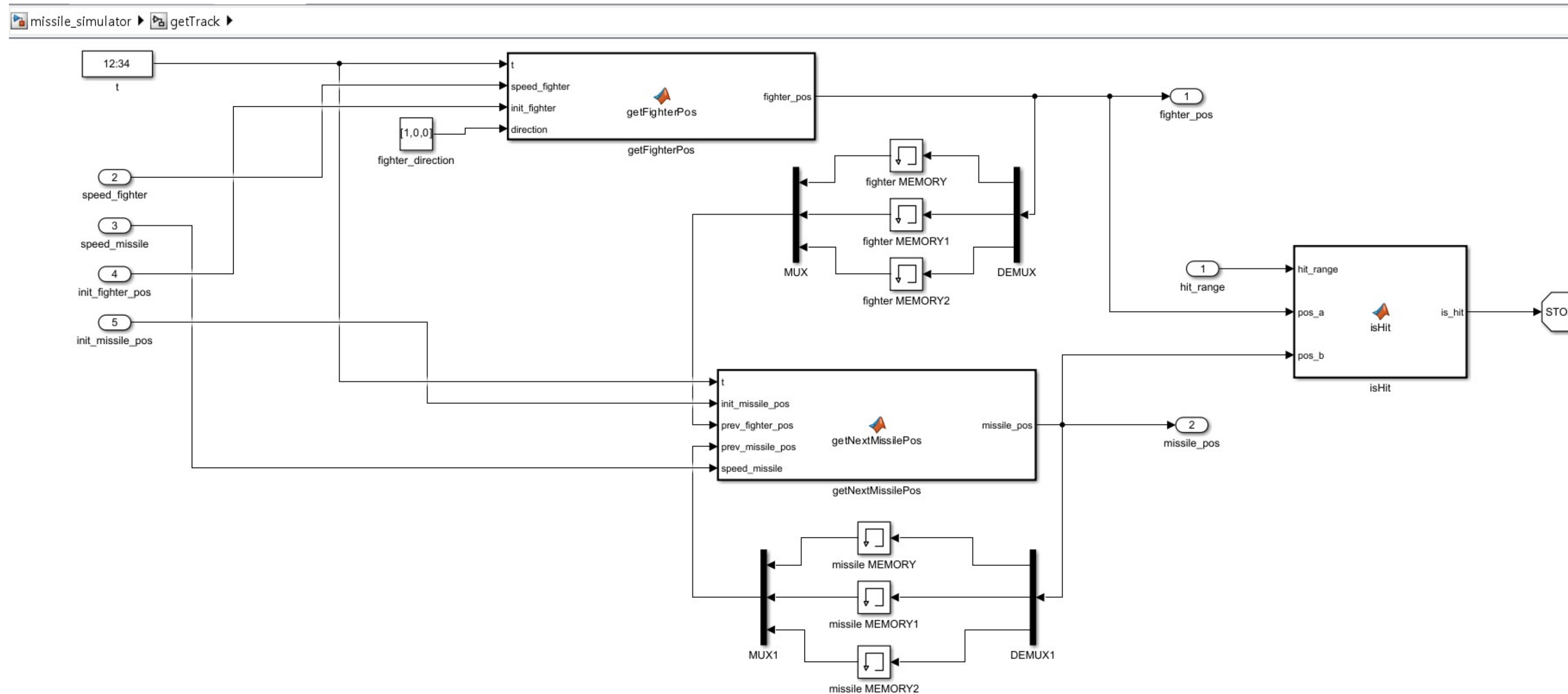
```
17
18
19 % 입력부:
20 time_limit = 1000;
21 hit_range = 5;
22 speed_fighter = 30;
23 speed_missile = 60;
24 init_fighter_pos = [0,0,100];
25 init_missile_pos = [500,0,0];
26
27
28 % 계산부:
29 % 시간 증분 10ms에 맞게 단위 조정
30 speed_fighter = speed_fighter/100;
31 speed_missile = speed_missile/100;
32
33 track_cell = getTrack(time_limit, hit_range, speed_fighter, speed_missile, init_fighter_pos, init_missile_pos);
34 %1: 종료시간, 2: 전투기항적, 3: 미사일 항적, 4: 명중여부 가 셀로 반환됨.
35
36 % 출력부:
37 % 그래프 출력
38 %track_cell - 1: 종료시간, 2: 전투기항적, 3: 미사일 항적, 4: 명중여부
39 displayPlot(track_cell{1}, track_cell{2}, track_cell{3}, track_cell{4});
40
41
42 % 명중여부 콘솔 출력
43 is_hit = track_cell{4};
44 if (is_hit)
45     disp("Target Hit !!");
46     fprintf("Eliminated time : %.2f (sec)\n", track_cell{1}/100)
47 else
48     disp("miss!");
49 end
```

```
1 % speed_missile %미사일 속력, 단위: / 10ms
2 % init_fighter_pos %전투기 초기좌표 (x,y,z), 단위 : km
3 % init_missile_pos %미사일 초기좌표 (x,y,z), 단위 : km
4
5 % output var :
6 % track_cell      %(1x4 cell) (1: 항적종료시간, 2: 전투기항적배열, 3: 미사일항적배열, 4: 명중여부)
7 % function :
8 % getFighterPos()
9 % getMissileDirection()
10 % getNextMissilePos()
11 % isHit()
12
13 % parameter :
14 % tmp_fighter_track %전투기 항적 저장 임시 배열 [x1, y1, z1; x2,y2,z2; ...]
15 % tmp_missile_track %미사일 항적 저장 임시 배열 [x1, y1, z1; x2,y2,z2; ...]
16 % direction %미사일 진행방향을 나타내는 단위벡터 [x,y,z]
17 % t %시간값을 지정하기 위한 임시변수
18
19 function track_cell = getTrack(time_limit, hit_range, speed_fighter, speed_missile, init_fighter_pos, init_missile_pos)
20
21 % 입력부:
22 % 전투기/미사일 항적행렬 초기값 지정 (t=1)
23 tmp_fighter_track(1, :) = init_fighter_pos;
24 tmp_missile_track(1, :) = init_missile_pos;
25
26
27 % 계산부:
28 % t값 초기화, 미사일 방향 단위벡터 선언
29 t = 2;
30 direction = [];
31 while (t <= time_limit)
32     tmp_fighter_track(t, :) = getFighterPos(t, speed_fighter, init_fighter_pos); %전투기 좌표 계산 함수
33
34     prev_f_pos = tmp_fighter_track(t-1, :); %t-1 에서의 전투기 좌표
35     prev_m_pos = tmp_missile_track(t-1, :); %t-1 에서의 미사일 좌표
36     direction = getMissileDirection(prev_f_pos, prev_m_pos); %미사일 진행방향 단위벡터 계산 함수
37
38     tmp_missile_track(t, :) = getNextMissilePos(prev_m_pos, direction, speed_missile); %미사일 좌표 계산 함수
39
40
41     % 명중하거나 or 시간 초과되면 항적 기록 종료
42     if (isHit(hit_range, tmp_fighter_track(t,:), tmp_missile_track(t,:)) | t>=time_limit)
43         break;
44     end
45
46     t = t+1;
47 end
48
49 % 출력부:
50 track_cell{1} = {t}; %종료시간
51 track_cell{2} = {tmp_fighter_track}; %전투기 항적 행렬
52 track_cell{3} = {tmp_missile_track}; %미사일 항적 행렬
53 track_cell{4} = {isHit(hit_range, tmp_fighter_track(t,:), tmp_missile_track(t,:))}; %명중 여부
54
55
56 end
```

- Class-Diagram 기반 각 모듈 개발
- getFighterPos(), getMissileDirection()
- getNextMissilePos(), isHit(), displayPlot()

# Development Process

## Simulink

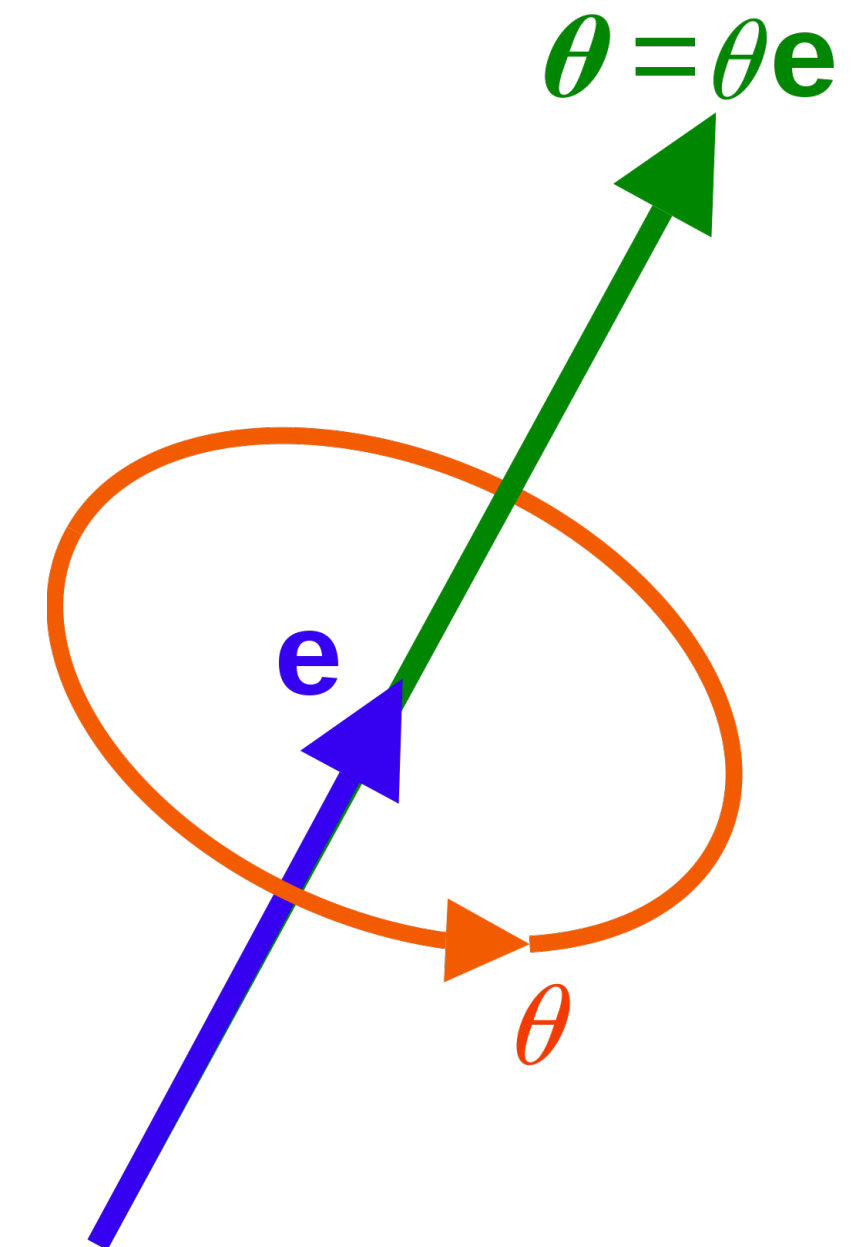
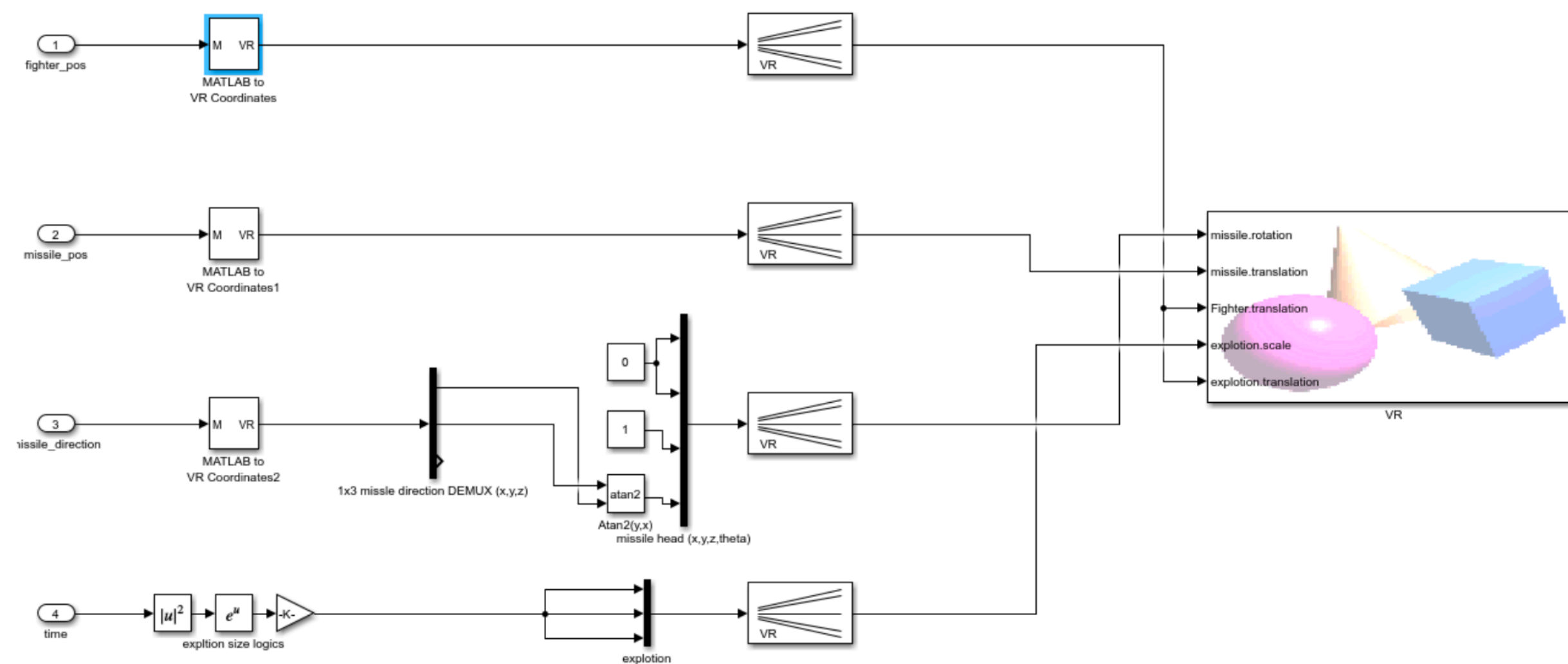


- Time-based Simulation
- Subsystem, User-Defined Function



# Development Process

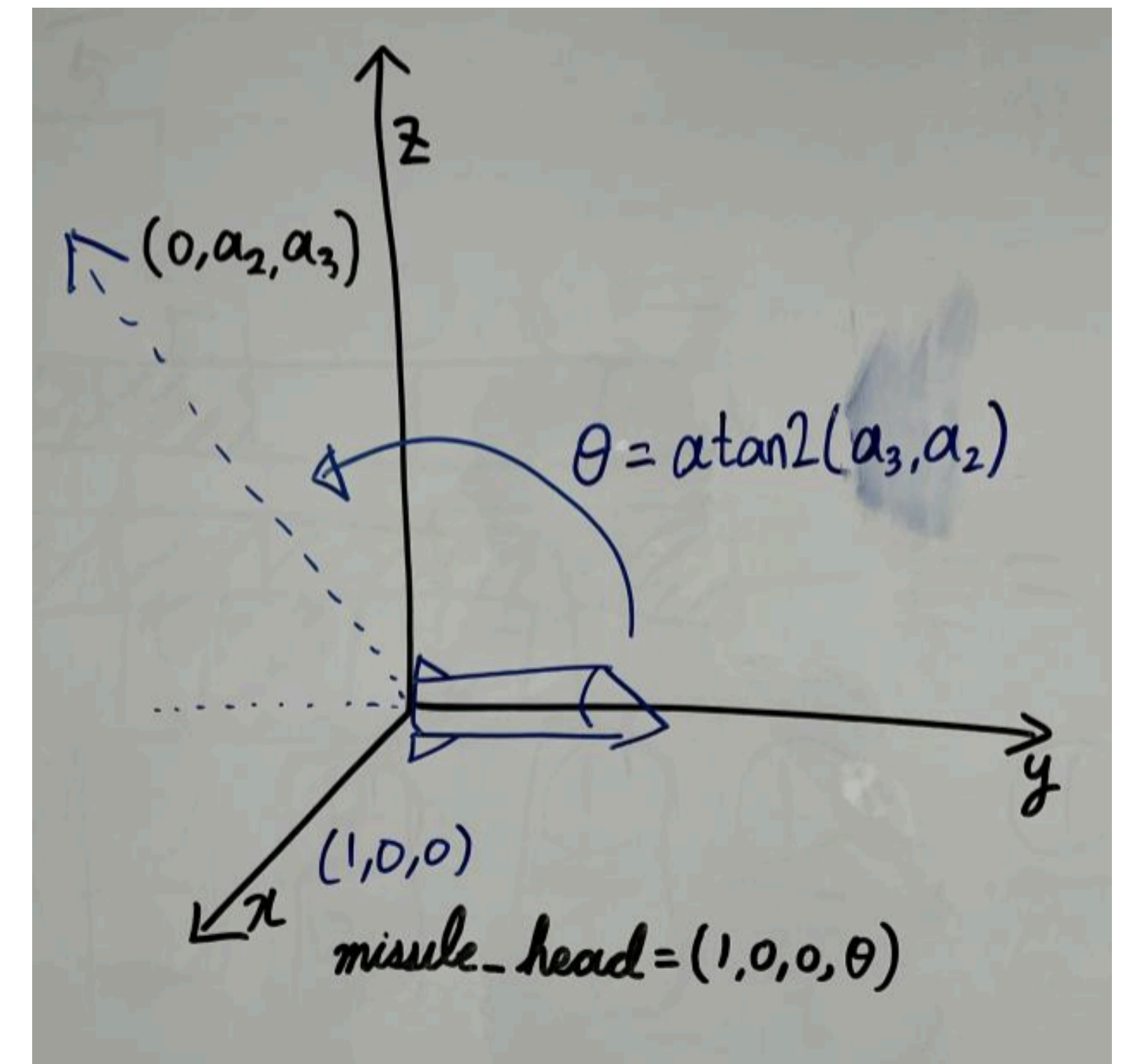
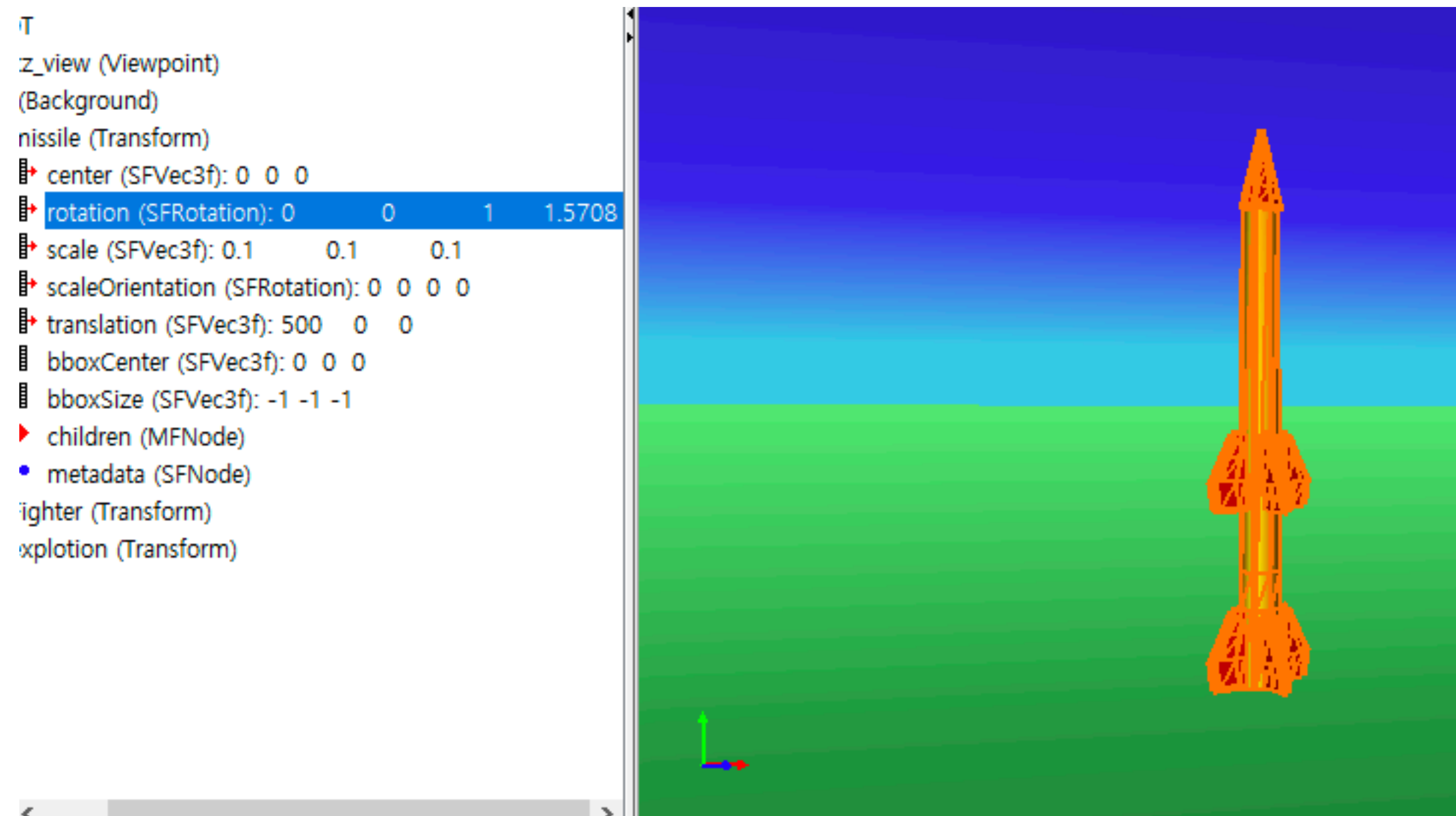
## 3D VR Visualization



- MATLAB to VR Coordinates (x,z,-y)
- Hit effect
- Missile head direction? (axis-angle representation)

# Development Process

## 3D VR Visualization



- MATLAB to VR Coordinates (x,z,-y)
- Hit effect
- Missile head direction? (axis-angle representation)

# Demonstration



Simulink : [https://youtu.be/jm-nV\\_AdHyc](https://youtu.be/jm-nV_AdHyc)

# Thank you!

## Q&A

Contact:

- [qorckddls010@gmail.com](mailto:qorckddls010@gmail.com)
- <https://github.com/Changin>