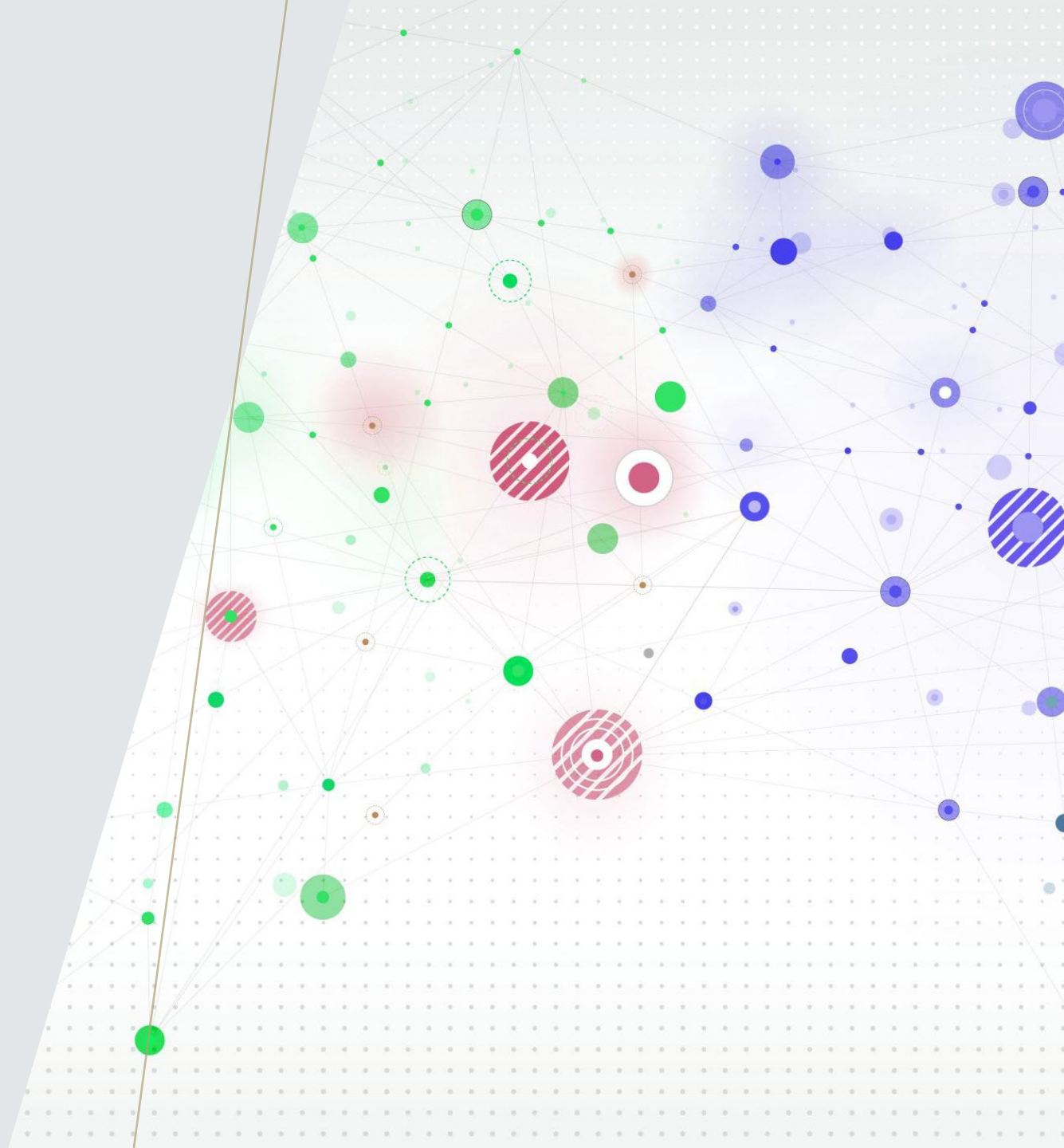


# 아주나이스

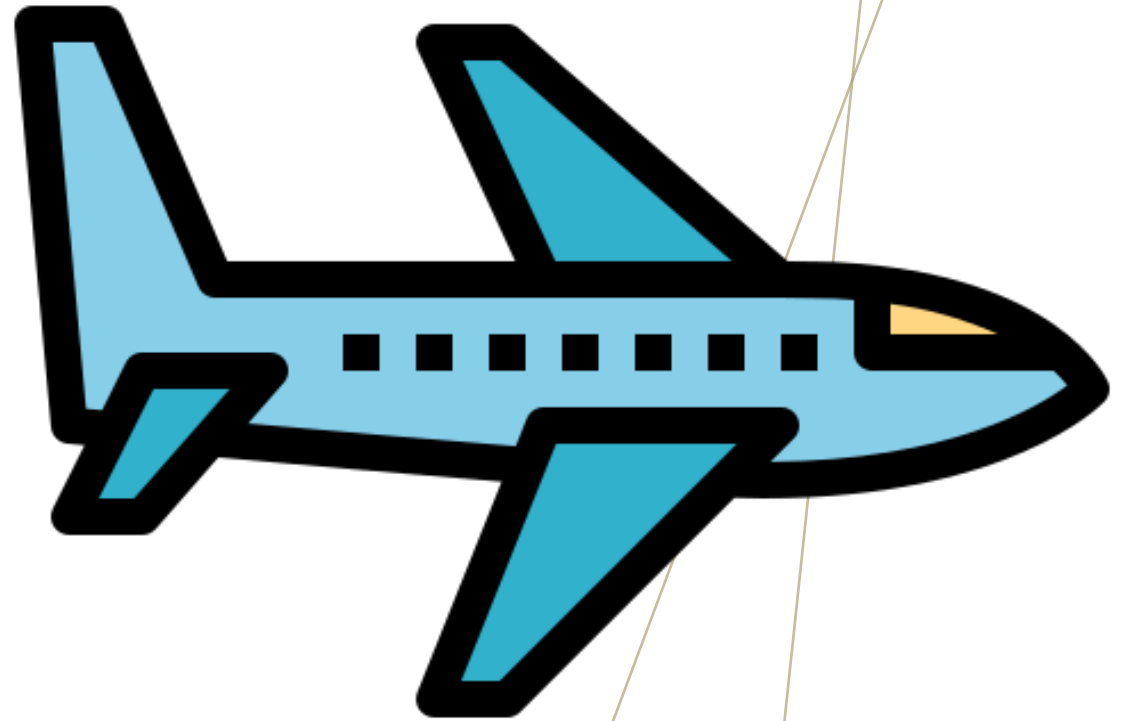
박준상 정유민 신준화

지도교수: 조광제 교수님



# 주제

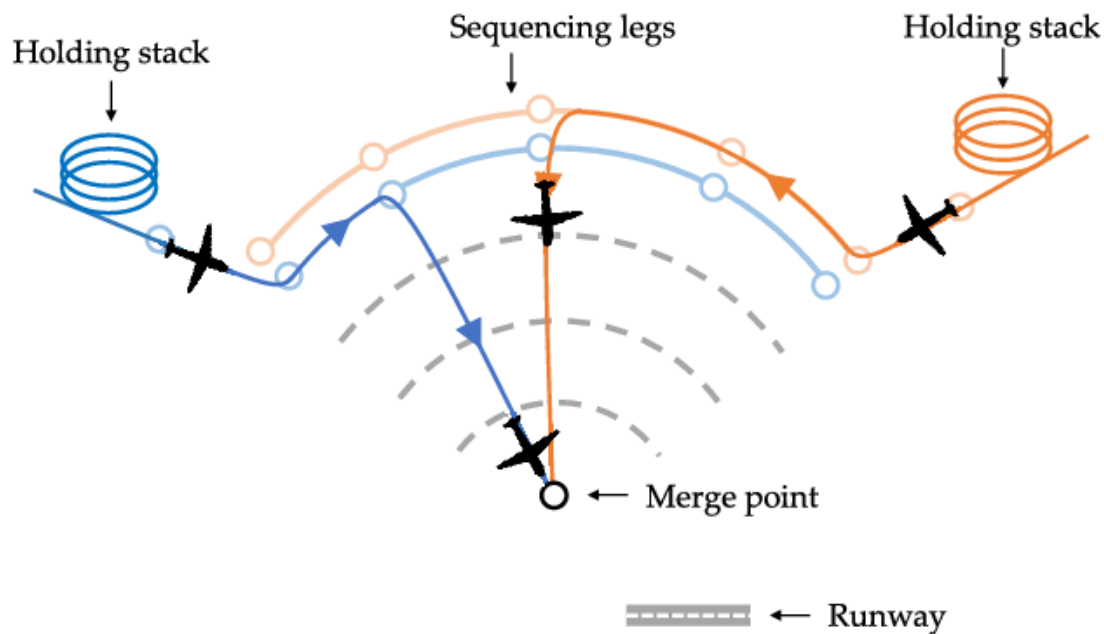
- 비행기 착륙 ATM(항공교통관리)  
알고리즘 개발 및 검증



# 도전목표

- ❖ Point Merge 기법 기반
- ❖ 다수의 항공기가 하나의 활주로에 착륙하고자 할 때 가장 효율적으로 착륙할 수 있는 알고리즘 개발
- ❖ 시뮬레이터(UNITY)에 표현 → 알고리즘 검증 및 효율성, 장단점 분석

# Point Merge 기법



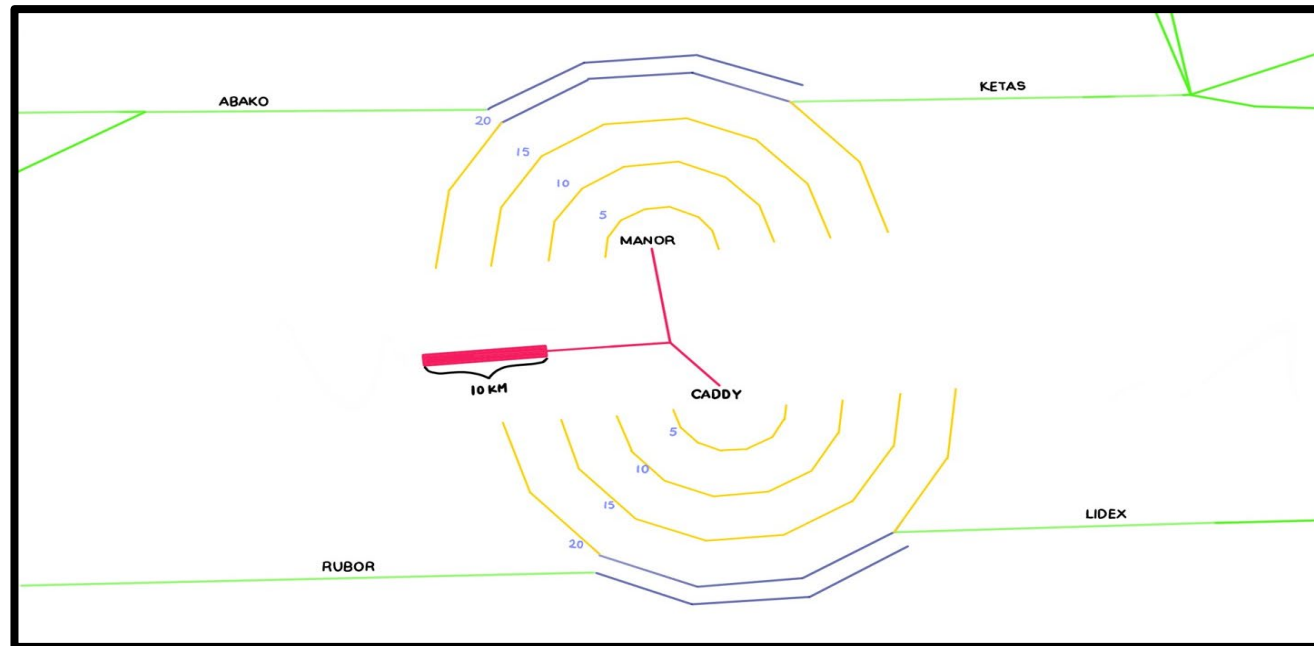
- ❖ **Merge point** = 활주로 착륙 시작 지점
- ❖ Merge point로부터 거리가 같은 점들 찍어 부채꼴 만들
  - → 부채꼴의 호 = **Sequencing legs**
  - → Sequencing legs에 찍혀 있는 점 = **Merge Fix**
- ❖ 항공기는 분리시간이 충족되지 않을 시 화살표 방향으로 이동하며 Merge Fix를 차례로 지남
- ❖ 처음 만난 Merge Fix에서 Merge point로 가는 경로에 다른 항공기 있다면 그 Merge Fix 지나친 후 다음 Merge Fix로 선회

# 연구 방법

- I. 태블릿 이용하여 Point Merge 기법이 적용될 배경 제작
- II. UNITY를 이용 → 항공기 이동속도, 고도 설정 및 이동
- III. 항공기 클릭 시 항공기 고도, 속도, 편명 나오는 팝업창 제작
- IV. 항공기에 Point Merge 기법 적용
- V. 시뮬레이터 테스트 및 피드백 & 수정
- VI. 현재 공군에서 이용하는 착륙 방법인 Holding Point 방법과 비교 분석 및 장단점 파악

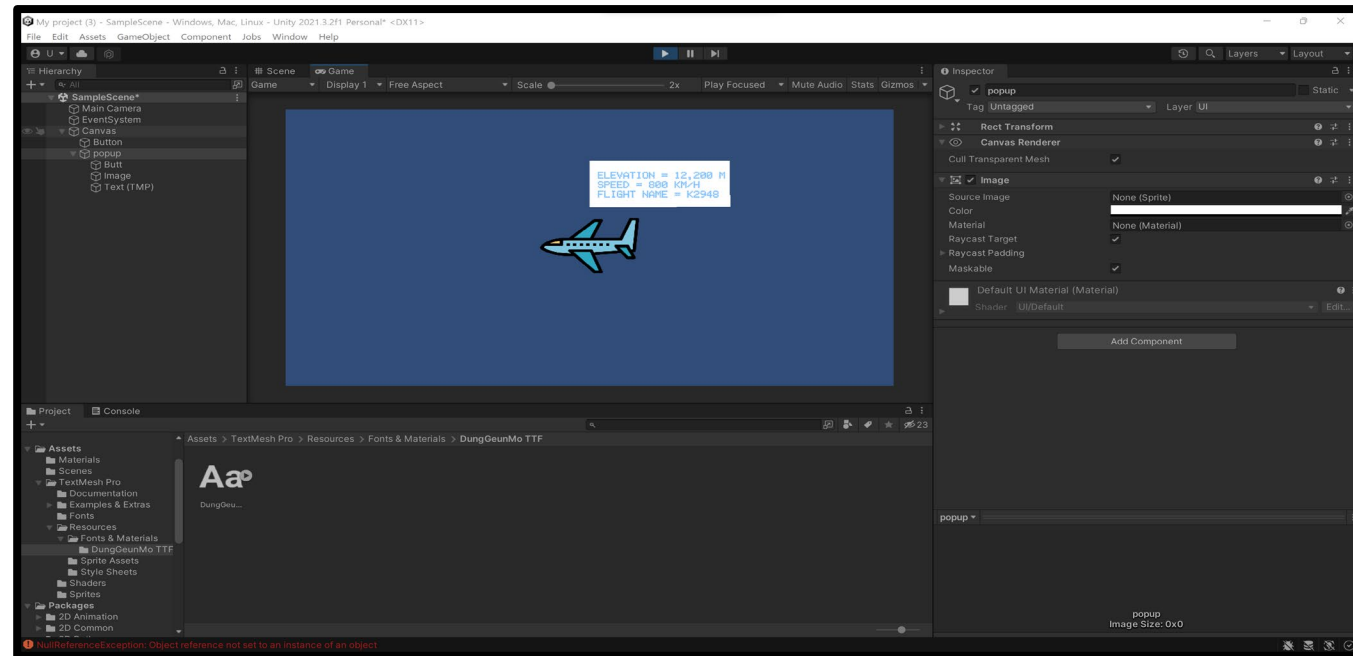
# 연구 진행(1)

1. UNITY를 이용 → 항공기 이동속도, 고도 설정 및 이동



# 연구 진행(2)

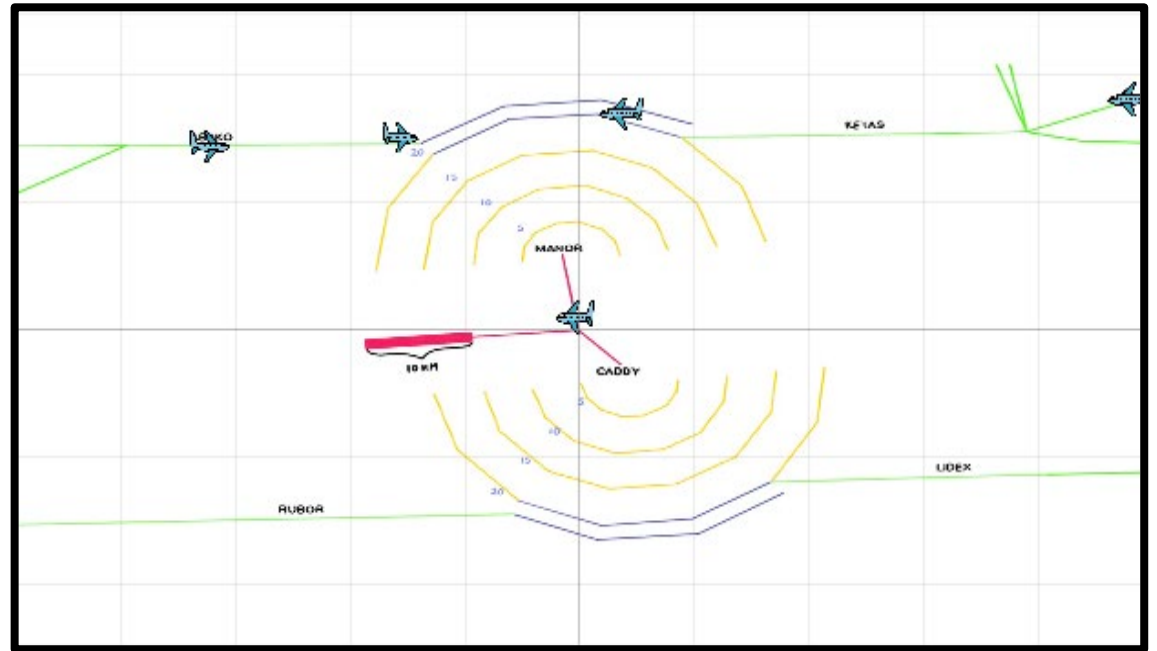
## 2. 항공기 클릭 시 항공기 고도, 속도, 편명 나오는 팝업창 제작



# 연구 진행(3)

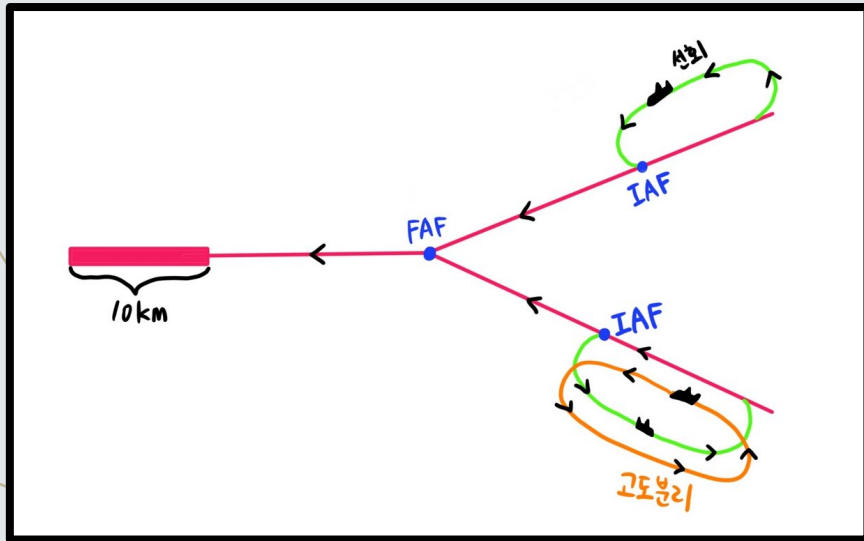
## 3. 항공기에 Point Merge 기법 적용

```
5 public class move : MonoBehaviour
6 {
7     public Vector3[] waypoints; //이동포인트 배열
8     public Vector3 currPosition; //현재위치
9     public int waypointIndex = 0; //이동포인트 인덱스
10    public float speed = 15f; //속도
11    //public float ETA_2_1 = 30.18033 / speed ;
12
13
14    // Start is called before the first frame update
15    void Start()
16    {
17        waypoints = new Vector3[7];
18
19        //이동포인트 값 할당
20        waypoints.SetValue(new Vector3(34, 25, 0), 0);
21        waypoints.SetValue(new Vector3(39, 15, 0), 1);
22        waypoints.SetValue(new Vector3(10, 15, 0), 2);
23        waypoints.SetValue(new Vector3(-1, 6, 0), 3);
24        waypoints.SetValue(new Vector3(0, 1, 0), 4);
25        waypoints.SetValue(new Vector3(-3, 0, 0), 5);
26        waypoints.SetValue(new Vector3(-15, -1, 0), 6);
27
28        //공통 거리: 19
29        //거리: 11.1803398875
30
31    }
32 }
```





# 최종 연구 목표



현재 공군에서 이용하는  
Holding Point 착륙 방식

## • <Holding Point>

- ❖ 고도의 활용 → 항공기가 착륙 의사 표시 시 홀딩포인트로 들어감 → 분리시간 확보
- ❖ 관제 방식 단순 → 관제사의 실수로 인한 사고 방지 → 공군 본부에서 이용
- ❖ 활주로로 착륙 시작하는 FAF도 단순함을 위해 홀딩포인트 둘 중 한 곳에 붙여둔다.

## • <Merge Point>

- ❖ 관제 복잡 & 고도의 이점 사용 불가
- ❖ But. 모든 방향에서 오늘 항공기들이 굳이 2개로 한정되어 있는 홀딩포인트까지 올 필요 X → 빠르게 선회 시작