

시뮬레이션의 활용을 통한 교내 드론 배송시스템 구축

**Building an on-campus drone delivery system  
through the use of simulation**

날들

국방디지털융합학과

# 목 차

---

1 연구배경

2 연구주제 및 성과물

3 배우거나 느낀 점

4 결론

---

# 연구 배경

# 연구 배경

---

- 국내 드론 배송시스템이 활성화 되어있지 않은 이유
  - 드론의 물리적 한계
  - 법적 규제에 한계
- 시스템을 구축하고 있다면 빠르게 상용화 될 수 있음
- 향후 공군 기지내 군수 물자 수송시스템 구축에도 도움이 될 수 있음

---

# 연구 주제 및 성과물

# 단계별 목표

---

## ■ 1학기

아주대학교의 모델링 및 수요도, 거리적 이득을 가중치로 고려한 드론 착륙지 선정 알고리즘이라는 두가지 목표를 종합해 최종적으로 시뮬레이션에 쓰일 맵을 구현

## ■ 2학기

→ 1학기에 설계한 알고리즘과 A\*알고리즘을 3차원으로 확장하는 방법을 모색하려 함

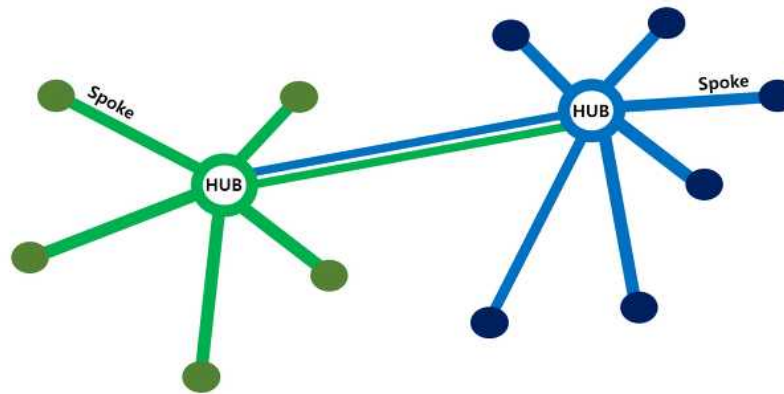
→ 그 다음 3차원으로 확장한 알고리즘을 Unity 3D내에 구현한 아주대학교 3D 모델링에 적용할 예정

---

# 성과물 1.

## ‘가중치를 적용한 최적의 **HUB**선정 알고리즘’

# 페르마 포인트란?

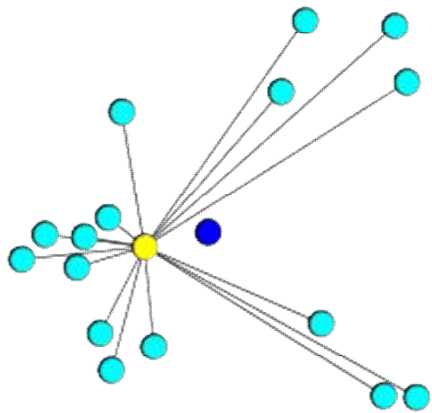


- 다각형 내부에서 각 꼭짓점으로 부터의 길이의 합이 최소가 되는 점  
→ 드론이 물건을 배송하는 과정에서 목적지를 순차적으로 경유하는 것이 아닌 목적지들에 대한 점을 HUB로 삼아 드론이 HUB에 물건을 전달하면 사람이 HUB에 이동하여 직접 수령



# 기하중점이란?

---



## *Geometric Median (기하중점)*

유클리드 공간에서 각 점들에서 거리의 합이 최소가 되는 점

→ 본 연구에서 추구하는 여러 수요지에서의 HUB지점

**BUT** 기하 중점을 다루는 법이 기하적으로 증명 X

→ Weiszfeld's Algorithm을 사용하여 근사

# 가중치를 적용한 이유

---

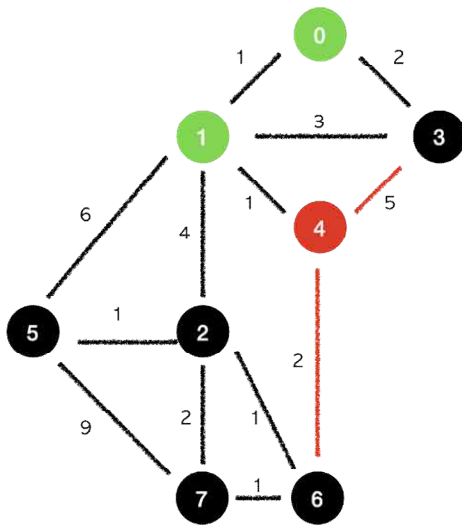


## 가중치?

- 배송을 기다리는 사람의 수
- 고객의 특성
- 배송 물건의 특성

→ 목적지마다 중요도가 다를 수 있기 때문에 가중치 개념을 HUB 선정 알고리즘에 적용

# 가중치를 적용한 기하중점



## 가중치를 적용한 기하중점

- 가중치 \* (기하중점에서 해당 좌표까지의 거리)의 합이 최소가 되는 점을 채택
- Weiszfeld's Algorithm을 변형하여 사용

**BUT** 예상과 다르게 가중치가 가장 큰 한 점이 기하중점이 되어버려서 새로운 방법을 모색함

# Weiszfeld Algorithm을 이용한 근사

$$W(\mathbf{x}^t) = \exp_{\mathbf{x}^t} \left( \frac{\sum_{i=1}^k w_i^t \log_{\mathbf{x}^t}(\mathbf{y}_i)}{\sum_{i=1}^k w_i^t} \right) \quad \text{if } \mathbf{x}^t \notin \mathcal{Y}$$
$$= \mathbf{y}_j \quad \text{if } \mathbf{x}^t = \mathbf{y}_j$$

where

$$w_i^t = d(\mathbf{x}^t, \mathbf{y}_i)^{q-2}.$$

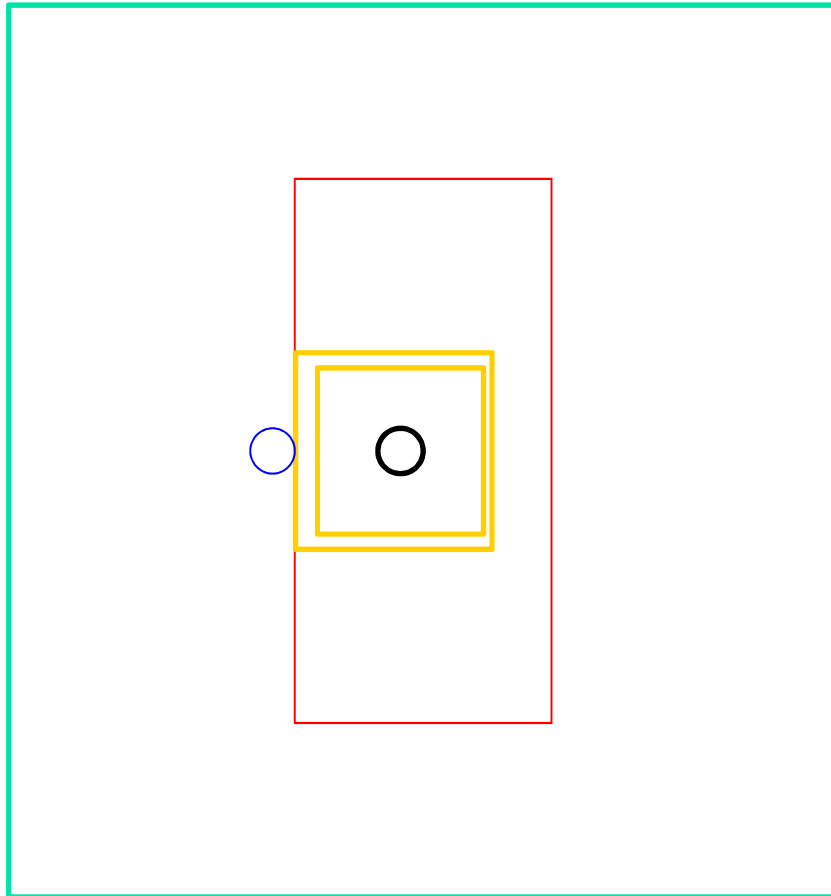
## *Weiszfeld Algorithm을 이용한 근사*

→ 기존의 방식대로 거리\*가중치의 합으로 알고리즘을 설계하면 기대와 다른 결과가 도출됨

→ 이에 기존 알고리즘에서 거리에 가중치를 곱하여 리턴한 후, 합의 최소가 아닌 거리\*가중치의 최댓값의 최솟값을 찾으려 알고리즘을 변경함

# 드론이 착륙할 수 없는 곳이 **HUB**라면?

---

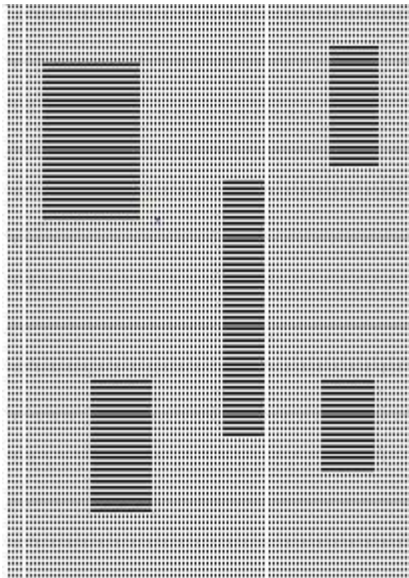


빨간색은 장애물  
검은색 원은 1차 도출된 HUB  
파란색 원은 최종 도출된 HUB

노란색 네모크기를 키우며 판단

# 성과물 1

## 가중치를 고려한 드론 착륙지 (이하 HUB라 칭함) 선정 알고리즘 결과

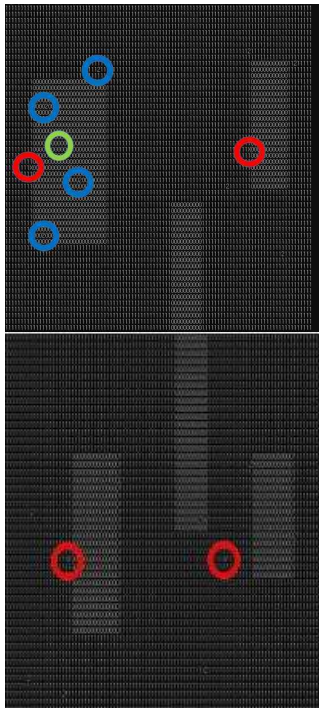


[0과 1로 표현한 가상의 맵]  
(진하게 보이는 부분이 드론이 착륙할 수 없는 곳)

- 가상의 맵을 토대로 상황에 따라 적절한 HUB가 도출 되는지 확인
- 통상적인 경우인 *Case 1* 을 기준으로 비교해 보면 결과를 알 수 있음

# 성과물 1

## Case 1. 통상적인 경우/가중치 모두 동일



2 : 드론을 요청한 지점(파랑)

X : 드론이 착륙할 지점(빨강)

결과:

1구역 HUB : 26, 8

2구역 HUB : 79, 19

3구역 HUB : 23, 79

4구역 HUB : 79, 69

※ 연두색 원의 위치에 착륙해야 하나 장애물이 있기에 빨간색 원에 착륙하였다

※ 나머지 세 빨간색 원은 같은 과정을 거침

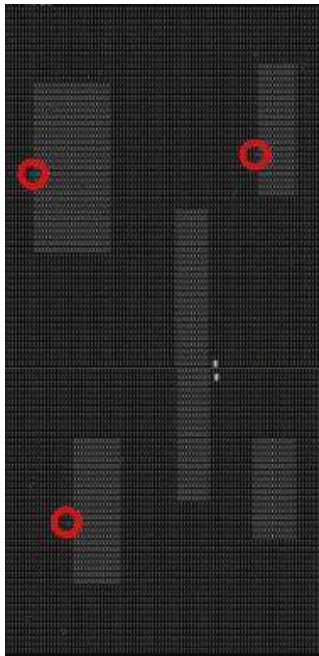
※ 이때의 파란색, 연두색 원은 생략함

※ 좌상단이 1구역, 우상단이 2구역, 좌하단이 3구역, 우하단이 4구역

# 성과물 1

---

Case 2. 한구역에서 드론을 요청한 사람이 아무도 없는 경우 (4구역 예시)/가중치 모두 동일



결과:

1구역 HUB : 26, 8

2구역 HUB : 79, 19

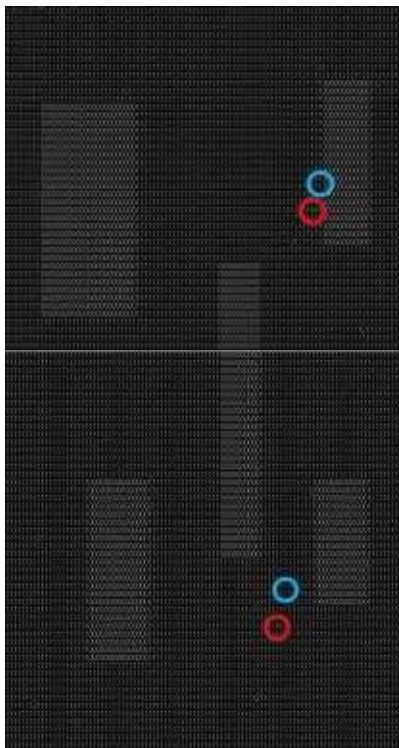
3구역 HUB : 23, 79

4구역 HUB : 없음



# 성과물 1

## Case 3. 통상적인 경우/각 구역마다 가중치 1~3 랜덤 부여



결과:

1구역 HUB : 23, 8  
2구역 HUB : 84, 20  
3구역 HUB : 25, 77  
4구역 HUB : 84, 68

비교군 : 가중치가 모두 1로 같은 경우(Case 1)

1구역 HUB : 26, 8  
2구역 HUB : 79, 19  
3구역 HUB : 23, 79  
4구역 HUB : 79, 69

→ **최종 결과** : 가중치(수요도)에 따라 HUB의 위치가 변한다

※ 붉은 원은 가중치가 바뀐 후의 HUB의 위치, 푸른 원은 Case 1 일 때의 HUB이다

※ 1,3구역은 따로 원으로 표현하지 않았으나 존재함

---

## 성과물 2.

### ‘UNITY 내 교내 캠퍼스 맵 제작’

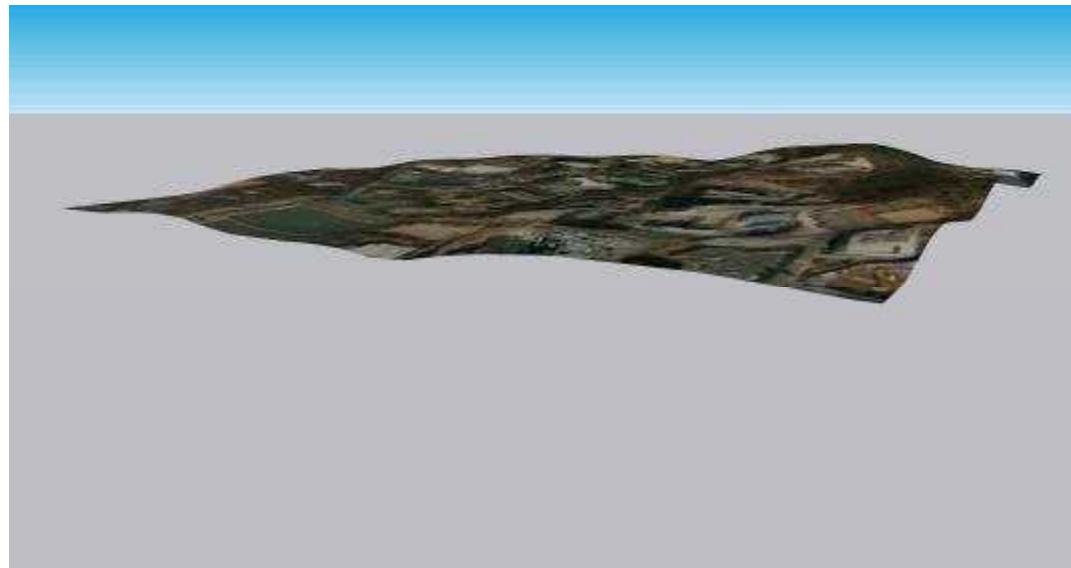
## 성과물 2

*Unity 3D* 내에 아주대학교 지형 및 건물을 맵으로 모델링

- 구글 스케치업과 *Unity*를 활용해 2D 지형을 3D 지형으로 변형



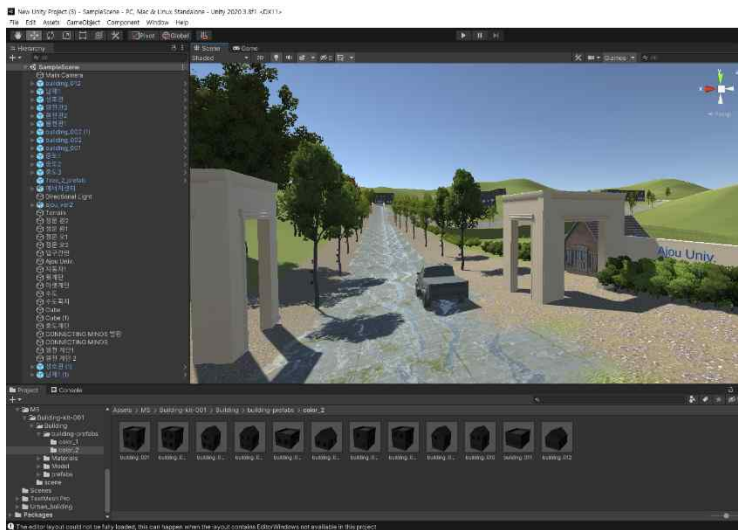
아주대학교 캠퍼스의 2D 캡처  
- Google Earth 활용



스케치업으로 지형의 실제 높낮이를 고려하여 3D 형태로 변형

# 성과물 2

## Unity 내 아주대학교 캠퍼스의 구현 (3D)



아주대학교 정문을 모델링하는 과정



아주대학교 울곡관 및 Connecting Minds 전경

---

## 느낀 점 및 결론

# 배우거나 느낀 점

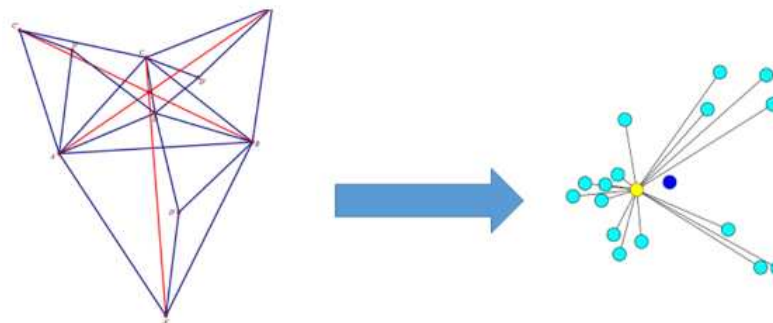
## 1. 페르마 포인트를 이용한 알고리즘 개발 : 연구 방향의 변경

처음에는 삼각형의 페르마 포인트를 응용하여 다각형에서의 페르마 포인트를 활용하려고 했으나 새로운 개념을 수학적으로 증명하는 것은 굉장히 어려운 일이었다.

이에 굴하지 않고 조사를 계속하던 끝에 ‘기하 중점’이라는 개념을 발견하였고 이는 우리가 찾던 개념과 완전히 일치하는 것이었다.

기하 중점을 구하는 방법은 수학적으로 증명되지 않았지만 이를 추정하는 알고리즘을 발견하였고,

이를 통해 이상적인 상황을 가정하는 문제 풀이와는 다르게 현실세계에서는 오차범위 내에서 추정하는 것만으로도 문제해결에 큰 도움이 된다는 것을 깨달았다.



페르마 포인트의 확장  
정확한 증명

기하 중점  
대략적인 추정

# 배우거나 느낀 점

---

## 2. 통합 회피 알고리즘 개발(장애물 회피+격자+기하 중점) : 협업의 중요성 인식

기하 중점을 이용한 HUB 선정 알고리즘을 개발하니, 극단적인 상황의 경우 예외가 발생케 되었다. 이를 해결하기 위해 추가 알고리즘을 개발하고 만든 알고리즘들을 통합하기로 하였다. 이때 알고리즘 팀 내에서 역할을 나누어 수행하였는데, 구획을 나누는 알고리즘과 장애물 회피 HUB 선정 알고리즘을 각자 하나씩 맡아 개발하기로 하였다.

이에 역할을 나누어 알고리즘을 개발하면 시간과 노력이 많이 절약되는 것을 깨달았지만, 동시에 원활한 협력이 가능케 하려면 변수명이나 주석을 작성하는 방식 및 클래스를 선정하는 방식 등을 어느정도 통일해야 하기에 개발자들 사이에서 암묵적으로 지켜지는 규칙을 숙지하고 다른 사람들의 코드를 많이 학습하려는 노력이 필요할 것 같다.

# 배우거나 느낀 점

---

## 3. 유니티 매핑 진행 : 오브젝트간 충돌

실제로 유니티 내에 3D 맵핑을 진행하면서 지형과 오브젝트 및 오브젝트와 오브젝트 사이에서 실제와 다른 약간의 비틀림이 존재하였는데 이를 해결하는 과정에서 유니티는 컴포넌트 기반의 구조이기 때문에 오브젝트 간 의존성이 없어야 자연스러운 모양새가 만들어진다는 것을 깨달았다.

이에 Script Properties에 Event Listening을 할 오브젝트를 등록하는 식으로 오브젝트를 맵 상에 구현한다면 컴포넌트 간의 의존성을 느슨하게 만들 수 있다는 사실을 배웠다.



---

# 감사합니다