

AR 기반 실내외 통합 길찾기 서비스

요약

기존 길찾기 애플리케이션은 2D 기반으로 하며 실내의 경우 정확도가 낮은 서비스를 제공하고 실외의 경우도 사용자가 추가로 주변을 탐색해야 하는 번거로움이 존재한다. 본 문서는 AR 기술을 기반으로 한 실내외 통합 길찾기 서비스를 제공하여 사용자가 직관적이고 정확하게 정보를 찾을 수 있는 길찾기 애플리케이션을 제안하고 구현한다.

1. 서론

1.1. 연구 배경

최근 코로나 팬데믹이 종료되면서 사람들의 외부 활동이 증가하였고 이러한 활동을 보조해주는 애플리케이션이 많이 개발되고 있다. 그리고 AR 기술의 발전으로 산업, 교육, 게임 등 여러 분야에서 AR이 활용되고 있다.

외부 활동을 지원해주는 AR 어플리케이션도 현재 많이 상용화 되어있다. 대표적으로 포켓몬이란 가상의 생물을 현실 세계에서 만날 수 있는 포켓몬 고(Pokemon Go)와 여러 지도 어플리케이션에서 지원하는 AR 길찾기서비스 등이 있다.

현재 많은 길찾기 애플리케이션은 보편적으로 2D를 이용한 서비스를 제공하고 있다. 또한 실내에서의 길찾기 서비스를 지원하지 않고 AR 서비스는 외부에서만 지원한다. 외부에서 실내의 특정 지점까지 가는 경로를 AR로 요청을 하면 실내로 들어가는 입구까지만 지원해주고 정작 실내에서의 경로는 지원하지 않는다.

기존 길찾기 어플리케이션과 다르게 사용자가 실내외 어디에 있던 AR 콘텐츠를 제공하여 지도에 대한 사용자 경험을 풍부하게 하도록 구현하였다.

사용자가 출발지/목적지를 실외로 정한다면 GPS 좌표계를 이용하여 위치를 특정한다. 실내로 정한다면 서버에 저장된 정보를 이용한다. 서버는 실내의 특정 거리마다 위치 및 주변의 스캔 정보를

담은 앵커를 저장한다. AR 길찾기를 요청하면 주행 유도선 또는 화살표 등의 AR 콘텐츠를 띄워서 사용자에게 안내해주는 UI를 제공한다.

또한 P2P 통신으로 시스템의 알고리즘에 따른 경로가 아닌 상대방이 직접 길을 안내해주는 기능을 추가하여 개인이 알고있는 가장 효율적인 경로를 공유하는 기능을 제공한다.

실외뿐만 아니라 실내에서도 AR 서비스를 지원해주는 길찾기 이므로 단순 2D 지도보다 원활하게 사용자가 목적지를 찾을 수 있다.

1.2. 연구 목표

AR 기술을 활용하면 사용자에게 직관적이고 정확한 정보를 줄 수 있다는 점과 현실 세계와 가상 세계의 결합을 통해 가상 콘텐츠와 상호작용하며 보다 흥미롭고 재미있는 활동을 즐길 수 있다는 장점을 활용한다.

첫 번째 목표로 길찾기의 최우선 목표인 정확한 길 안내를 위해 GPS 및 스마트폰의 가속도계와 자이로스코프를 이용해 외부의 위치 정보를 추적한다. 그리고 실내의 위치 정보 저장을 위해 실내에서 일정 거리의 Anchor마다 360도 파노라마 방식으로 카메라 스캔을 한다. 또한 실내에서 실외로 혹은 그 반대의 경우에서 UI가 자연스럽게 바뀌는 방법을 연구한다. 결과적으로 사용자가 실외와 실내에서 AR 기반의 길찾기 서비스를 제공받을 수 있도록 한다.

두 번째 목표로 WebRTC를 기반의 화면 스트리밍 기능을 지원하여 사용자끼리 직접 길을 안내할 수 있도록 한다. 터치를 하면 그 좌표가 상대방에게 전송되고 2D 좌표를 3D 좌표로 변환시키는 작업을 거친 후 AR 콘텐츠를 상대방의 화면에 띄울 수 있도록 한다.

2. 관련연구

2.1. AR 콘텐츠 개발

2.1.1. Unity

게임, 시뮬레이션 및 가상 콘텐츠를 포함한 2D 및 3D 애플리케이션을 만들 수 있는 게임 엔진 및 게임 플랫폼이다. 개발자가 복잡하고 동적인 대화형 콘텐츠를 만들 수 있도록 물리 시뮬레이션, 애니메이션 및 스크립팅 도구와 같은 다양한 기능을 제공한다. 또한 개발에 필요한 3D 모델, 텍스처, 애니메이션, 스크립트 등을 담은 리소스인 Unity Assets을 통해 프로젝트에 기능, 그래픽 및 기타 기능을 추가할 수 있다.

2.1.2. AR Foundation

Unity를 사용하여 AR 애플리케이션을 개발할 수 있도록 지원하는 프레임워크이다. ARCore와 ARKit을 모두 지원하며 Android와 iOS 기기 모두에서 실행되는 크로스 플랫폼 AR 콘텐츠를 만들 수 있다. 이미지 인식, 주변 환경 인식 및 모션 추적과 같은 AR 콘텐츠 제작에 필요한 통합 API를 제공한다.

2.1.3. ARCore, ARKit

ARCore은 Android 기기에서 AR 애플리케이션을 구축하기 위한 Google 플랫폼이고 ARKit은 iOS 기기에서 AR 애플리케이션을 구축하기 위한 Apple 플랫폼이다. 컴퓨터 비전, 모션 트래킹, light estimation 등의 기술 조합을 통해 현실 세계에 중첩될 수 있는 AR 콘텐츠를 제작할 수 있다. 또한 평면 감지 기술을 지원하여 이질감 없이 AR 콘텐츠가 화면에 나타나게할 수 있다. AR 콘텐츠를 디바이스 간에 공유할 수 있는 클라우드 기능도 제공한다.

2.2. 외부 위치 추적

2.2.1. GPS

GPS(Global Positioning System)는 지구상에 있는 어디에서나 위치와 시간 정보를 제공하는 위성 기반 네비게이션 시스템이다. 스마트폰의 GPS 수신기는 최소 4개의 위성으로부터 신호를 수신하고 이 신호를 사용하여 디바이스의 위치를 계산한다. GPS 시스템은 수 미터 이내의 정확성을 가지기 때문에 외부 위치 추적에 탁월한 기술이다. 그러나 건물이나 다른 구조물에 의해 방해를 받을 수 있는 실내에서는 정보의 신뢰성이 떨어진다.

2.2.2. 가속도계 및 자이로스코프

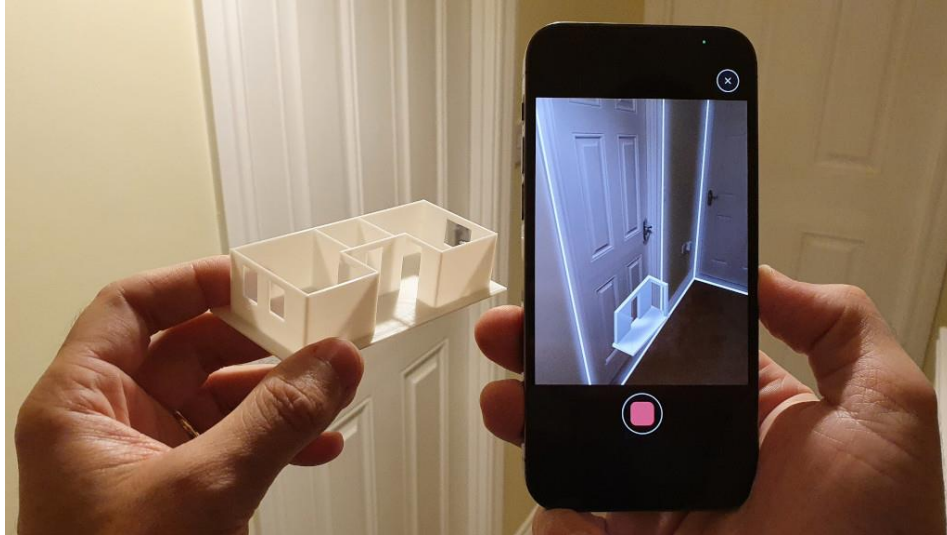
스마트폰을 포함한 전자 디바이스에서 발견되는 움직임과 방향을 측정하는 센서이다. 가속도계는 선형 가속도와 속도 변화를 측정하고 자이로스코프는 각속도와 방향 변화를 측정한다. 이 두 센서를 이용하여 3차원에서 장치의 방향과 움직임을 측정할 수 있다. GPS 정보와 결합하여 스마트폰의 움직임과 위치를 실시간으로 추적할 수 있다.

2.3. 실내 위치 저장

2.3.1. Anchors

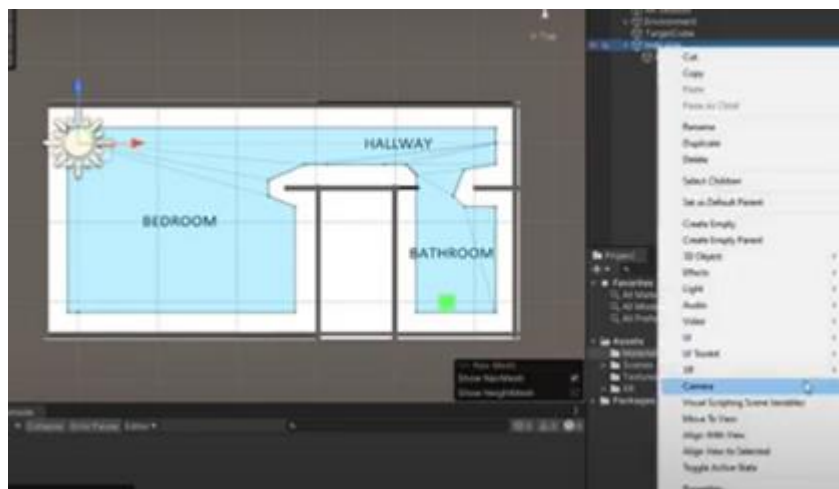
AR 추적을 위한 기준으로 사용되는 공간의 한 점이다. 한 Anchor에서 카메라 스캔을 하여 주변 정보를 저장하면 그 점에 대한 위치 정보를 저장할 수 있다. 가상 객체 또는 콘텐츠가 실제 환경에 올바르게 배치되도록 하는데 사용된다.

2.3.2. Polycam



iOS에서 작동되는 3D 스캐닝 앱이다. 스마트폰 또는 태블릿의 카메라를 사용하여 실제 객체나 공간을 스캔하고 3D 모델로 변환할 수 있게 해준다. 평범한 물체, 실내 및 실외 공간, 인물 등 다양한 대상을 스캔하여 obj 및 glb 파일로 만들 수 있다.

2.3.3. NavMesh



게임이나 시뮬레이션을 위한 네비게이션 mesh를 만들 수 있는 Unity의 기능이다. 시뮬레이션 환경에서 이동할 수 있는 영역을 나타내는 다각형을 설정할 수 있다. A* 알고리즘을 이용하여 두

지점 사이에서 가장 효율적인 경로를 제공해줌으로써 객체가 맵에서 현실적으로 이동할 수 있게 해준다. 장애물이 많거나 지형이 다양한 복잡한 환경에서 특히 유용하다.

2.4. 사용자 간의 데이터 교환

2.4.1 WebRTC

웹 브라우저 간에 플러그인의 동무 없이 서로 통신할 수 있도록 설계된 API이다. 별도의 소프트웨어 없이 비디오, 음성 및 일반 데이터가 피어간 실시간으로 전송 가능하게 해준다. 피어간 연결을 중계해주는 시그널링 서버에서 미디어의 정보를 나타내는 SDP와 네트워크 연결 정보를 담은 candidate를 교환해주면 이후 서버없이 P2P로 데이터를 송수신할 수 있다.

2.5. 기존 길찾기 애플리케이션 연구

현재 길찾기 애플리케이션은 보편적으로 2D를 이용하고 있어 사용자가 목적지 근처에 도달을 해도 직접 주변을 탐색하여 정확한 목적지가 어딘지 파악해야하는 번거로움이 있다. 뿐만 아니라 GPS 기술의 한계로 실내에서의 길찾기는 그 정확도가 낮다. AR 서비스는 외부에서만 지원하고 목적지가 실내라면 실내까지의 입구만 AR 서비스를 제공하고 있다.

2.6. 기존 연구의 문제점 및 해결 방안

2.6.1. 연구의 문제점

GPS 기술의 한계로 실내에서 길찾기 서비스를 제공하기 힘들고 실내와 실외에서 통합적으로 길찾기 서비스를 제공하는 상용화된 애플리케이션이 아직 없다. 이에 따라 처음 가보는 장소에서 지도 및 길찾기 서비스를 활용하여도 목적지를 찾기 쉽지않다.

2.6.2. 해결 방안

2.6.2.1. AR Cloud Anchor

실내의 건물 정보를 Vuforia Creator 및 Polycam으로 미리 스캔하고 특정 간격마다 Anchor을 설정하고 저장한다. 사용자가 스마트폰을 통해 서버로 전송하는 화면의 유사도로 현재 어느 Anchor에 있는지 판단하여 목적지로 가기 위한 다음 Anchor로의 경로를 디스플레이에 화살표 등의 AR 콘텐츠로 제공한다. 이를 통해 실외에서는 GPS로, 실내에서는 Anchor 기반으로 통합적인 길찾기 서비스를 제공한다.

2.6.2.2. 실내와 실외 길찾기 UI 따로 적용

외부에서는 GPS 기술 및 가속도계, 자이로스코프를 사용하고 실내에서는 앞서 언급한 Anchor 기반의 길찾기를 사용하기 위해 실내, 실외에서 각기 다른 길찾기 UI를 제공한다. UX를 떨어뜨리지 않도록 하는 UI를 전환 기술이 필요할 것이다.

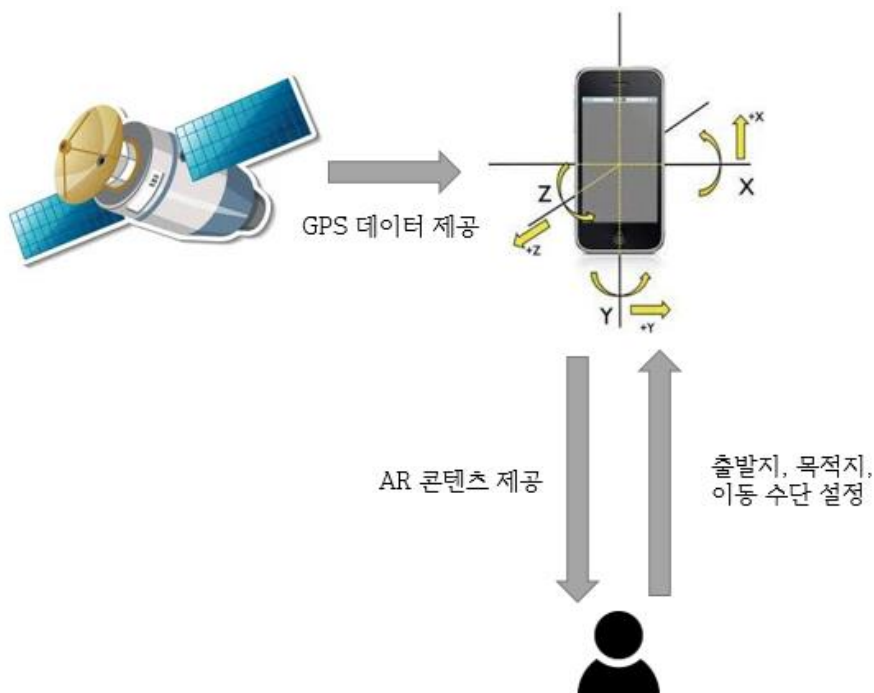
2.6.2.3 WebRTC를 통한 AR 콘텐츠 렌더링

화면 스트리밍을 통해 또다른 사용자가 화면을 보고 있다가 자신이 안내해주는 기능을 구현한다. 터치 이벤트를 WebRTC의 Data Channel을 통하여 전송하면 2D 좌표를 3D 좌표로 변환하는 알고리즘을 통해 적절한 위치에 AR 콘텐츠를 나타나게 하여 처음 가보는 장소에서도 정확한 목적지를 찾도록 한다.

3. 프로젝트 내용

3.1. 시나리오

3.1.1 실외



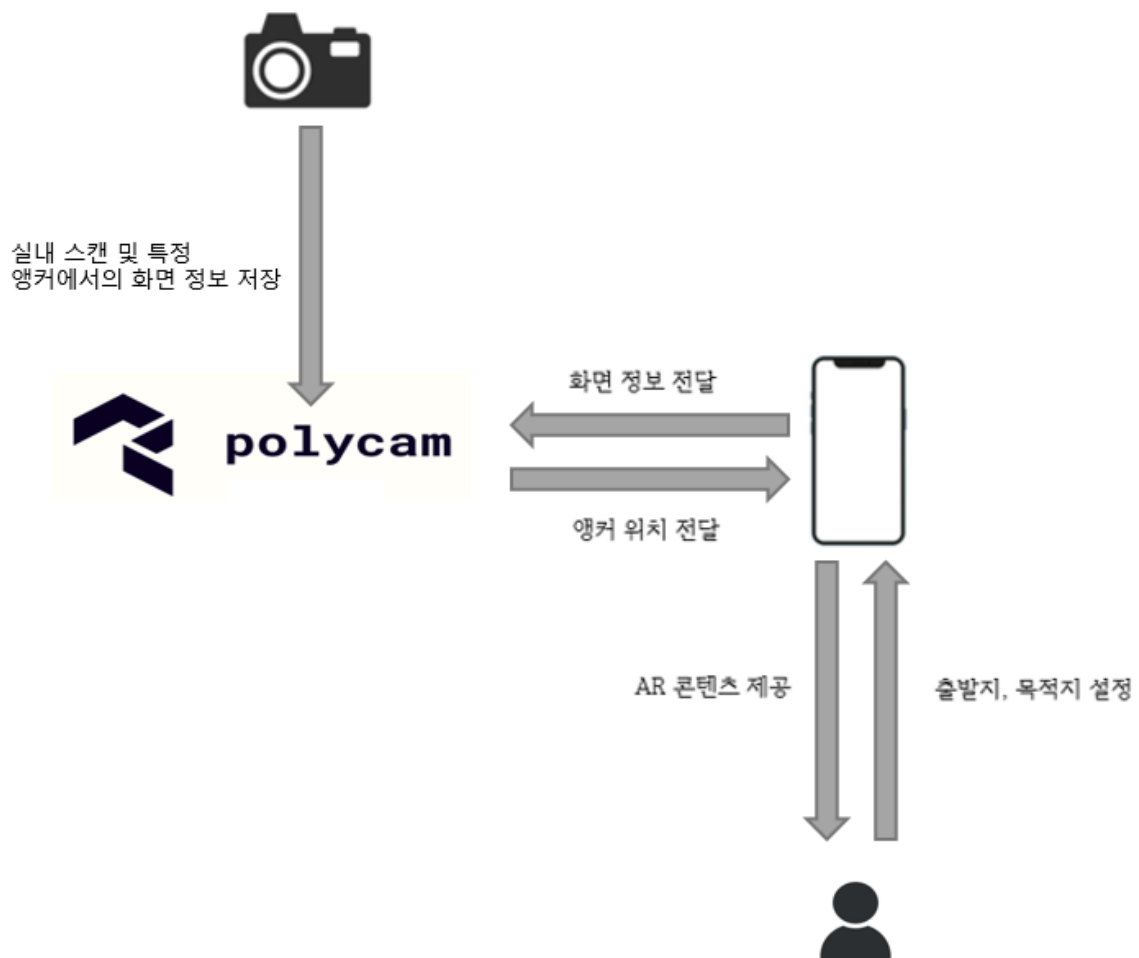
1. 사용자가 모바일에 출발지와 목적지, 이동수단을 입력한다.

2. GPS 좌표계 및 가속도계, 자이로스코프를 이용하여 사용자의 현재 위치를 파악한다.
3. 사용자의 현재 위치에서 도착지까지의 경로를 안내한다. 이때 길안내를 위한 도구로 화살표와 고속도로 유도선 방식을 이용한다.



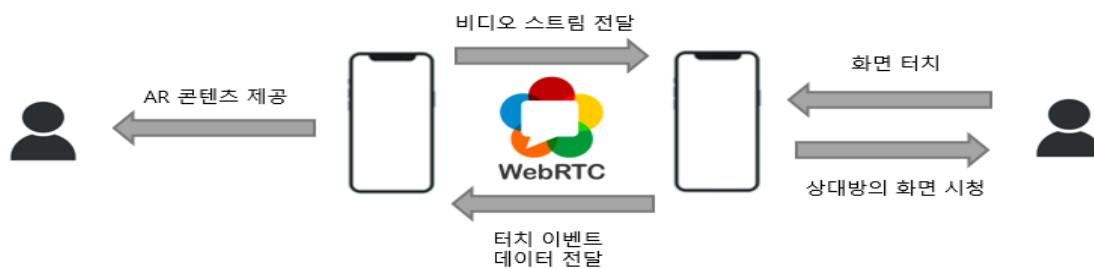
4. 도착지가 실내라면 실내의 시작점까지 실외 버전의 UI로 안내하도록 하고 실내 시작점에 도착하면 실내 길찾기 UI로 전환된다.

3.1.2. 실내



1. 사용자가 출발지와 목적지를 설정한다.
2. 사용자가 스마트폰 카메라로 주변을 비추면 클라우드에 저장된 정보와 유사도를 계산 후 현재 어떤 Anchor에 있는지 파악한다.
3. 최단 경로 알고리즘인 A* 알고리즘으로 목적지까지의 경로를 찾은 후, 사용자에게 현재 Anchor에서 다음 Anchor로 가는 경로를 안내한다.
4. 목적지 Anchor에 다르면 안내를 종료한다.

3.1.3. WebRTC



1. 사용자 A가 길찾기 서비스를 이용한다.
2. 다른 사용자 B가 WebRTC 연결 요청을 하고 A가 수락을 하면 A의 비디오 스트림이 B에게 전달되어 A의 화면을 본다.
3. B가 A에게 길을 안내하려고 B의 스마트폰 화면을 터치한다.
4. Data Channel을 통해 A의 스마트폰으로 터치 이벤트 데이터가 전달된다.
5. 2D 좌표에서 3D 좌표로 변환되는 알고리즘을 거친 후 AR 콘텐츠가 렌더링된다. 사용자 A는 B가 보낸 터치 데이터에 근거한 AR 콘텐츠를 보고 그 길로 이동한다.

3.2. 요구사항

3.2.1. 실외에서의 요구사항

- GPS 위치 정보에 디바이스의 가속도계, 자이로스코프가 담고있는 이전 위치와 움직임을 더한 값으로 정확한 위치를 구하도록한다.
- 사용자가 선택한 이동 수단에 따른 최적 경로를 제공한다. 실시간 교통 정보 API를 활용하여 사용자에게 추천 이동 수단 및 경로를 제공한다.

- 사용자가 직관적으로 길을 안내받을 수 있도록 고속도로의 유도선이나 화살표를 현실 세계에 중첩하여 나타나도록 구현한다.
- 길찾기 경로가 실외에서 실내라면 실내의 입구부터는 실내 버전 길찾기 UI로 전환한다. 전환 중 사용자의 UX를 떨어뜨리지 않도록하는 방법을 고안한다.
- 가속도계와 자이로스코프를 이용하여 이동 중 사고를 막기위해 사용자가 정지했을 때에만 콘텐츠를 제공하도록 한다.

3.2.2. 실내에서의 요구사항

- 경로가 실외에서 실내였다면 실내의 특정한 입구에도 앵커를 배치하고 주위 화면 정보를 저장한다.
 - Polycam 및 Vuforia Creator sdk를 활용하여 서버에 저장된 앵커 정보를 실시간으로 확인하도록 구현한다.
 - 층 간의 이동을 제공할 수 있게 3D map을 구현할 때 계단을 추가하여 z축을 다루는 방법을 구현한다.
 - 서버에 저장된 화면 정보와 현재 카메라가 보여주는 화면의 유사도를 판단하여 어떤 Anchor에 있는지 판단하는 알고리즘을 구현한다.
 - 이동중 카메라의 이동으로 Anchor 위치가 변하는 문제를 고려하여 실시간으로 Anchor의 위치를 동기화하는 알고리즘을 구현한다.

3.2.3. WebRTC 통신 요구사항

- 서비스를 이용하기 위해 회원 가입한 이메일을 키 값으로 사용한다.
- LTE 및 공용 Wifi을 쓰는 상황에서도 서비스를 이용할 수 있도록 사설 IP를 공인 IP로 변환해주는 서버인 TURN 서버를 구축한다.
- 시그널링 서버가 계속 가동될 수 있도록 AWS EC2를 이용하여 중단되지 않도록 서버를 배포한다.

3.3 시스템 설계

3.3.1 시스템 구성도

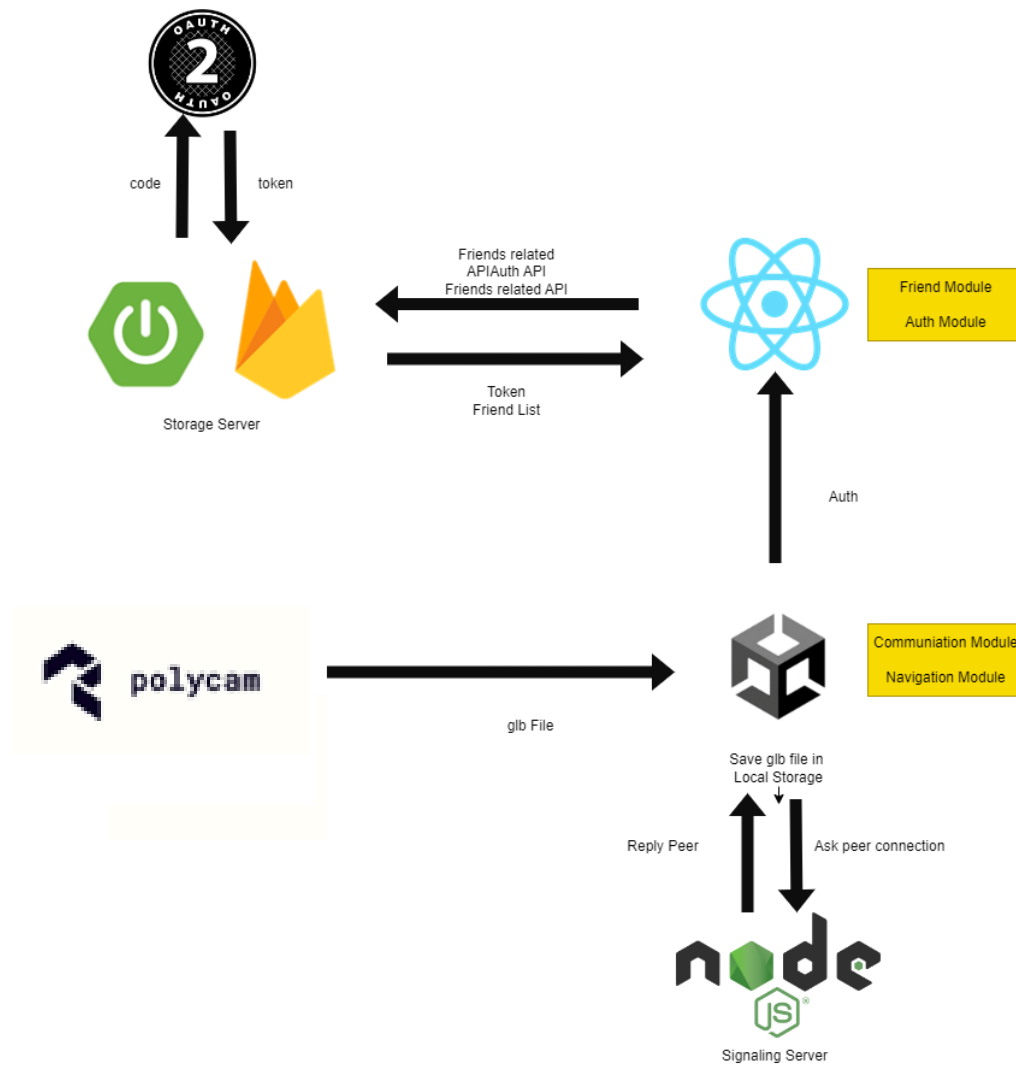


그림 1 - 시스템 구성도

시스템 구성도는 위 그림과 같다.

3.3.2. UML Diagram을 통한 시스템 모델링

3.3.2.1. Use Case Diagram

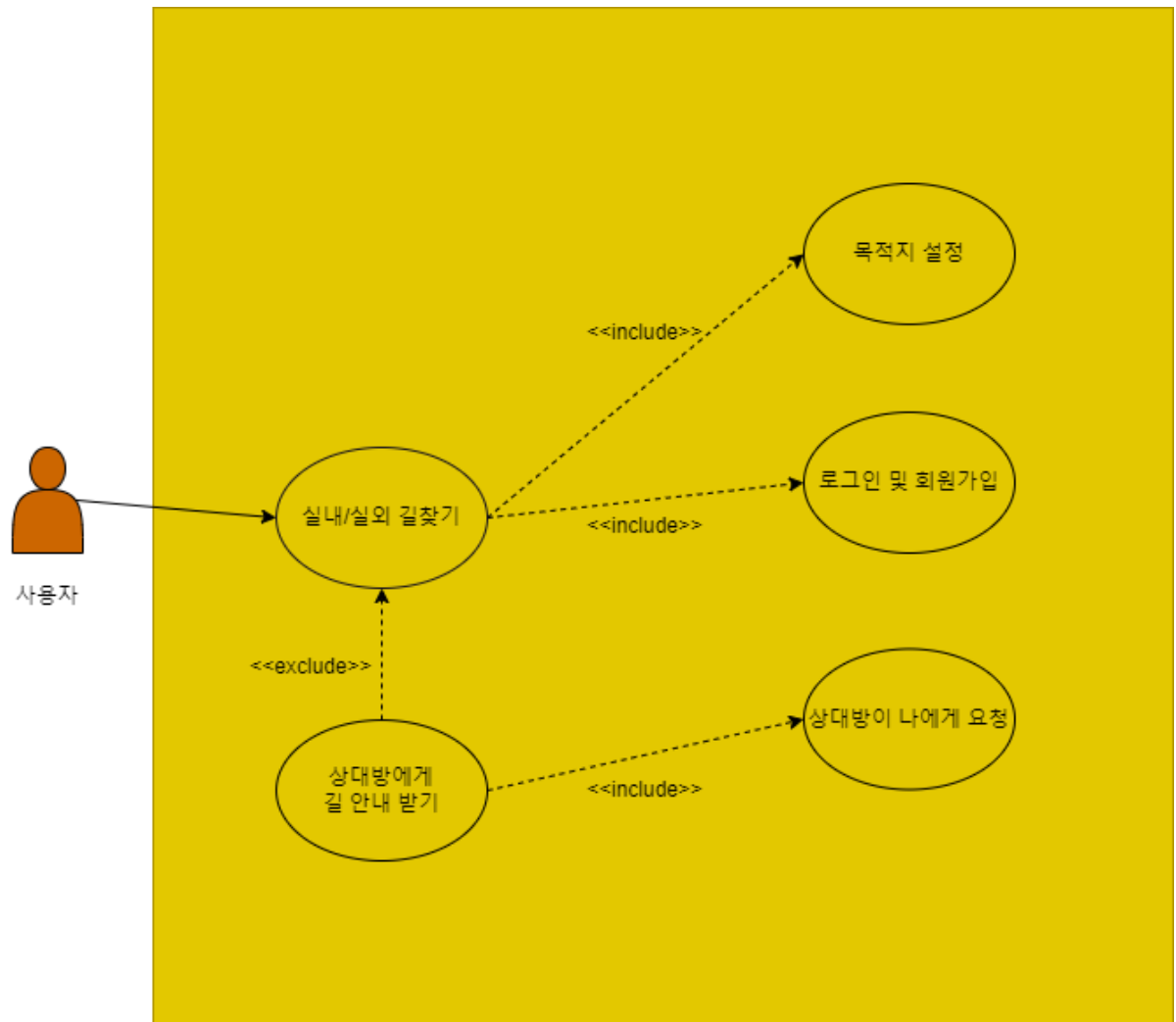


그림 2 - Use Case Diagram

사용자는 스마트폰을 이용하여 어플리케이션을 실행한 후 로그인과 목적지를 설정하면 실내 또는 실외 길찾기 서비스를 이용할 수 있다. 서비스를 이용하던 도중 상대방이 내 화면을 보는 것을 수락하면 화면이 스트리밍 되고, 상대방의 디바이스에서 터치를 하면 기존 길찾기는 종료되고 상대방에게 길을 안내받는다.

3.3.2.2. Class Diagram

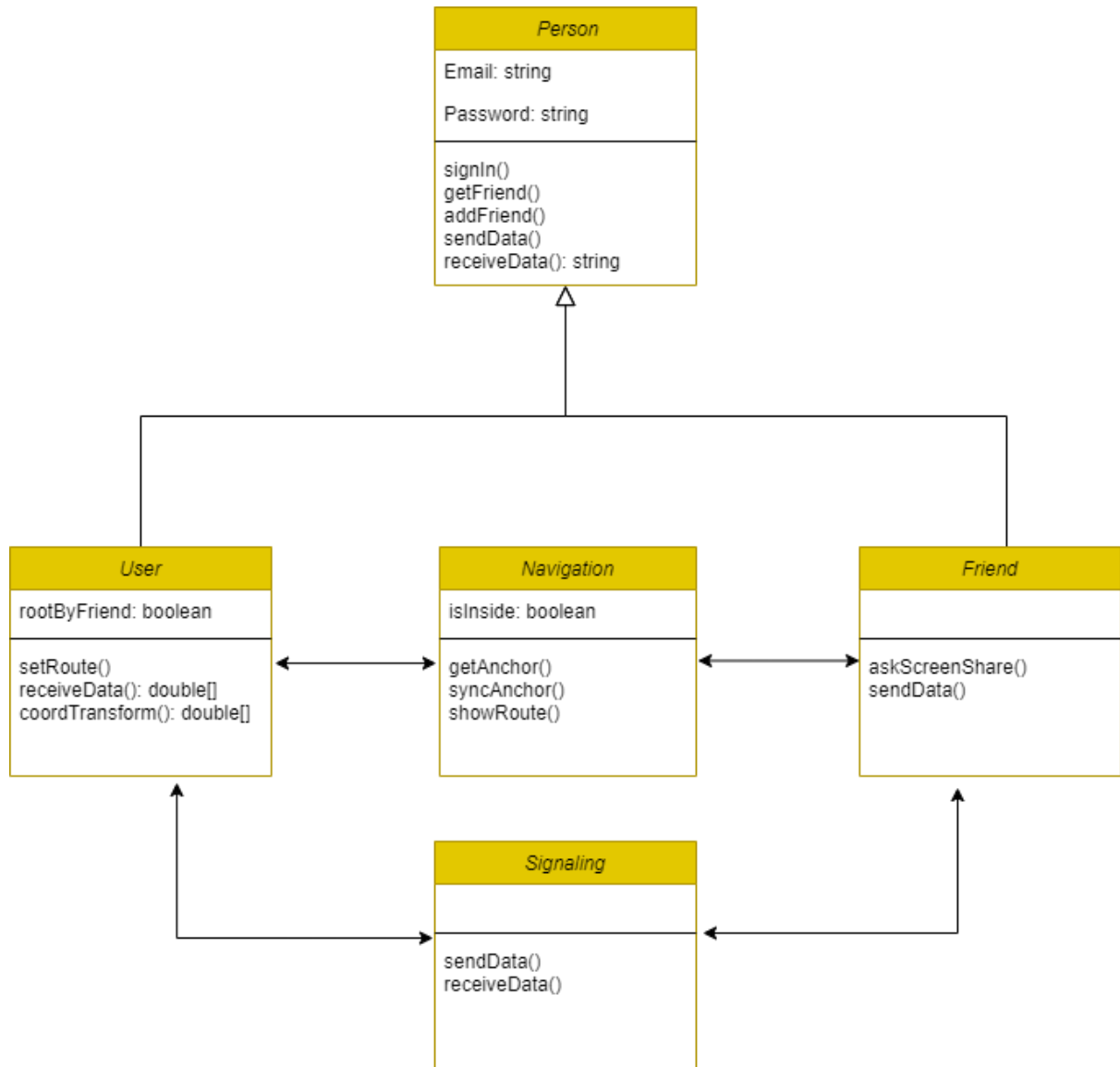


그림 3 - Class Diagram

클래스는 크게 User, Navigation, Friend, Signaling 부분으로 이루어져 있다. User 부분은 경로를 설정하여 Navigation으로 보내고 Navigation은 설정한 경로와 사용자의 실내인지 아닌지에 기반하여 최적 경로를 보여준다. Signaling 부분은 Friend가 User의 화면을 얻고자 요청하는 상황에서 네트워크 및 디바이스 데이터를 받아 상대방에게 전달한다.

3.3.2.3. Activity Diagram

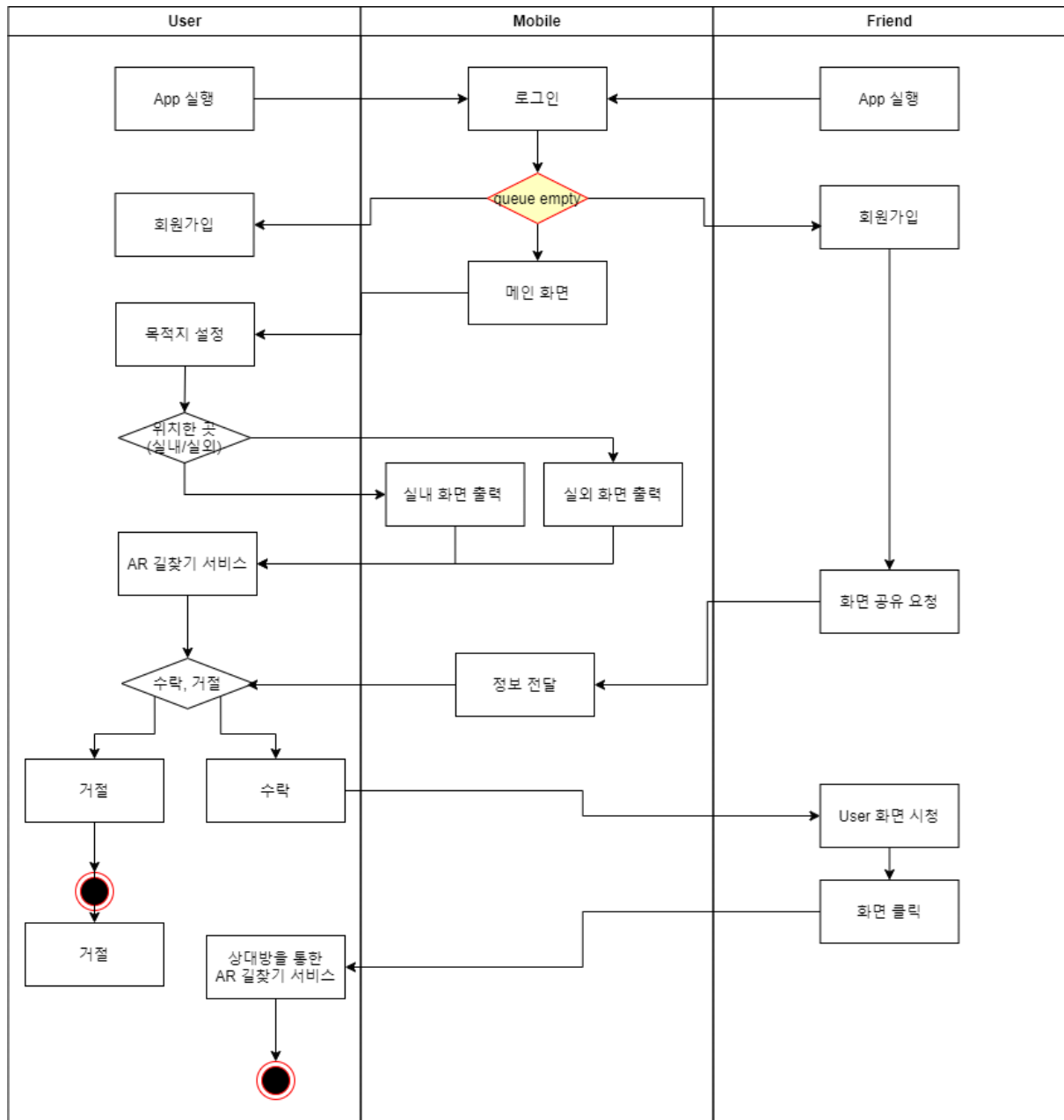


그림 4 - Activity Diagram

User은 어플리케이션을 실행하면 로그인 또는 회원가입을 하고 목적지를 설정하면 카메라를 통해 어플리케이션이 실내 또는 실외인지 확인한다. 그러면 User은 AR 길찾기 서비스를 이용한다. Friend의 경우 어플리케이션을 실행한 후 로그인을 마치면 User의 이메일 입력을 통한 화면 공유 요청을 한다. User이 수락하면 Friend는 User의 화면을 볼 수 있게 되고 화면을 클릭하면 그 지점에 AR 콘텐츠가 나타나게 되어 상대방이 길찾기 서비스를 제공할 수 있도록 한다.

3.3.2.4. Sequence Diagram

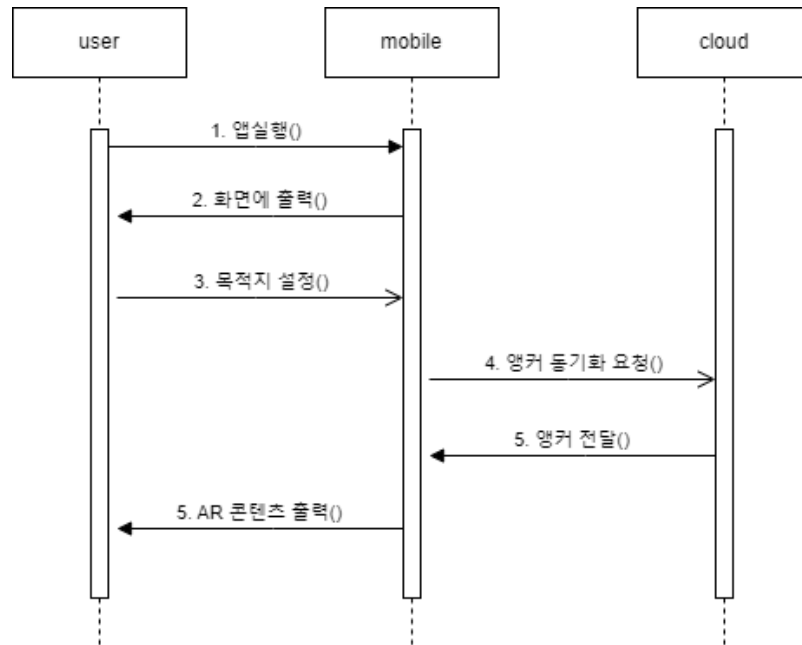


그림 5 - 실내 Navigation Sequence Diagram

실내 Navigation 기능의 Sequence Diagram을 보여주고 있다. 모바일 기기는 사용자가 목적지를 설정하기를 대기하고 있으며 입력이 들어오면 클라우드에 저장된 앵커를 불러오고 동기화를 하여 사용자의 위치를 파악한다. 그후 배치된 앵커에 기반하여 사용자에게 최적의 경로를 제시한다.

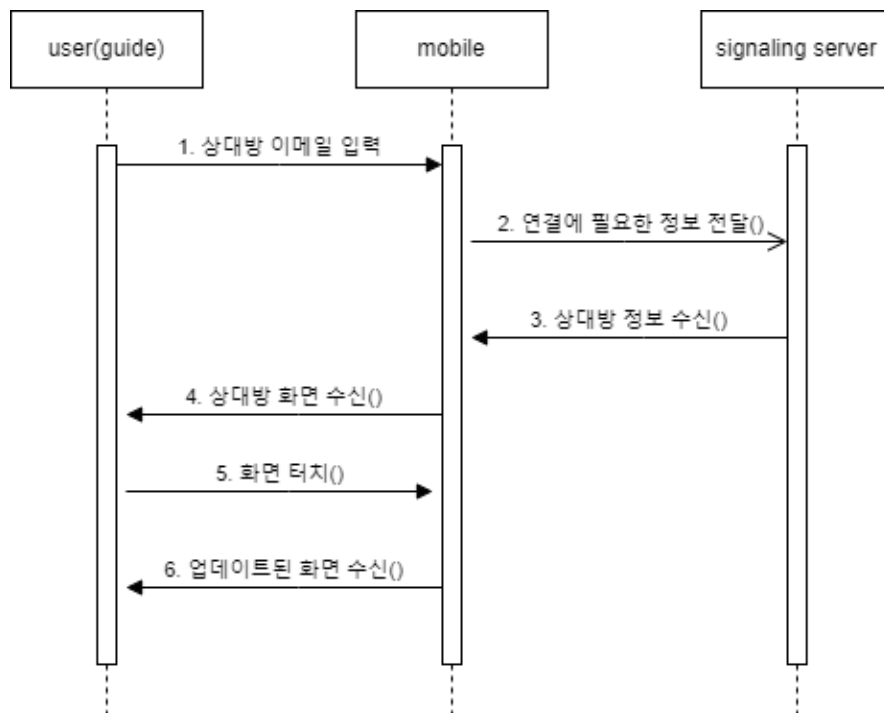


그림 6 - P2P Navigation Sequence Diagram

터치로 직접 길을 안내해주는 기능을 위한 Sequence Diagram을 보여주고 있다. 사용자는 상대방의 이메일을 입력하고, 모바일 기기는 서버에 이메일을 포함한 연결에 필요한 정보를 전달한다. 서버가 해당되는 유저에게 데이터를 보낸 후 받으면 그 정보를 모바일 기기가 다시 수신하여 사용자가 상대방의 화면을 볼 수 있도록 한다. 길을 안내해주기 위해 화면을 터치하면 모바일 기기는 상대방에게 바로 터치 데이터를 보내고, 업데이트된 상대방의 화면을 사용자에게 화면에 출력해준다.

3.4. 구현

Github URL(Application): <https://github.com/kugorang/CapDePlaylist>

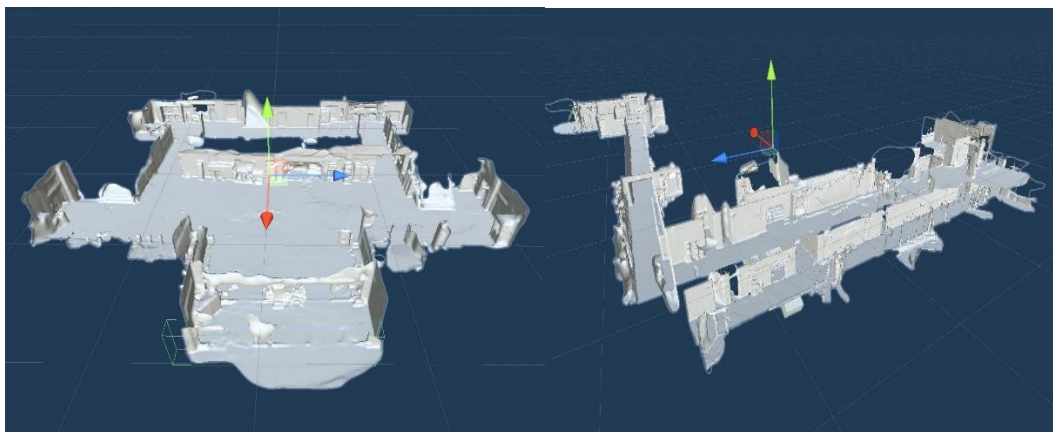
Github URL(Web): https://github.com/taemin4u/Cap_Web

4. 프로젝트 결과

4.1. 연구 결과

이 프로젝트의 진행에 가장 중요한 점은 클라우드에 저장된 AR Anchor를 이용하여 사용자가 실내 어디에 위치하는지 동기화하는 것이고 클릭 이벤트를 통해 전달된 2D 좌표를 3D 좌표로 변환한 후 그것에 기반한 위치에 AR 콘텐츠를 나타내는 것이다.

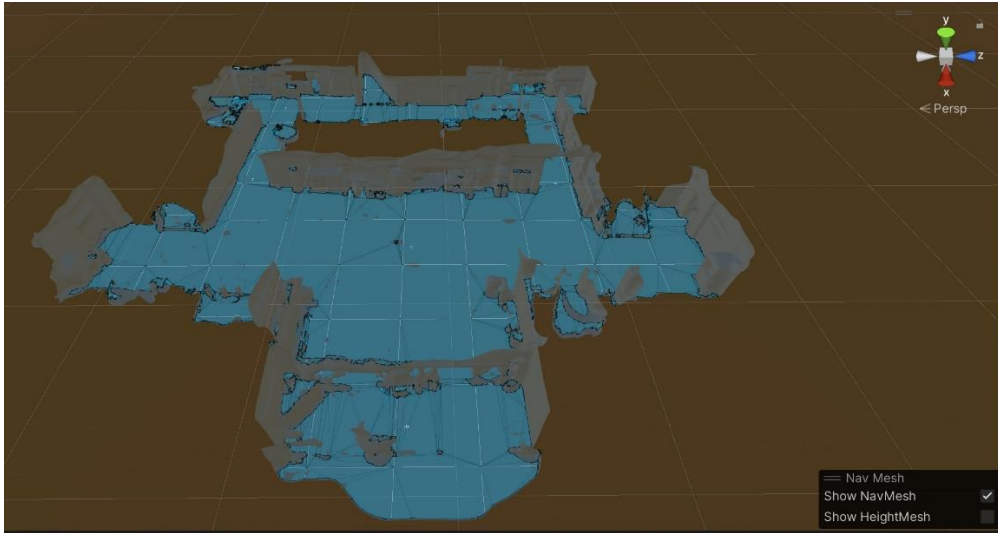
4.1.1. AR Cloud Anchor를 이용한 실내 네비게이션



[그림] 1

[그림] 2

먼저, 기능 개발을 위해 학생회관([그림1]), 전자정보대학 지하1층~1층([그림2])를 LiDAR센서가 있는 스마트 디바이스를 이용해서 3d scanning을 한다.



[그림] 3

그리고 실시간 경로검색을 위한 NavMesh를 Bake한다([그림3]).

이후, AR Space와 실제 space의 공간 좌표계를 맞추기 위한 AR Cloud Anchor를 건물바닥 구석구석에 세팅해놓는다.

최종적으로 AR디바이스가 AR Cloud Anchor를 인식할때마다 다음 코드를 통해서 3D Map의 Position,Rotation을 실제 환경에 맞게 동기화 한다.

```
MapsToAdjust[(int)mapType].transform.rotation = Quaternion.RotateTowards(
    AnchoredPointGameObjects[selectedIndex].transform.rotation,
    selectedCloudAnchorGameObject.transform.rotation,
    1000f
);
```

```
Vector3 vectorToMove = selectedCloudAnchorGameObject.transform.position -
    AnchoredPointGameObjects[selectedIndex].transform.position;//3D Map이 이동해야하는 벡터계산
MapsToAdjust[(int)mapType].transform.position += vectorToMove;
```

selectedIndex : 현재 선택된 3D Map Index

MapsToAdjust : 3D Map들의 배열

AnchoredPointGameObjects : 3D Map상의 Anchor위치에 위치한 동기화용 게임오브젝트

selectedCloudAnchorGameObject : 현재 color camera에 인식된 AR Anchor 오브젝트

4.1.2. 2D 좌표의 3D 좌표 변환

디바이스마다 해상도가 각기 다르므로 클릭 이벤트가 발생 2D 좌표를 상대 좌표로 변환하여 전송한다. 시스템 내부에서 상대 좌표를 절대 좌표로 변환한 후 Unity Raycast를 이용하여 올바른 곳에 AR 콘텐츠가 렌더링되도록 한다.

```
private void Start()
{
    InitializeDirections();
    StartCoroutine(CheckComplete());

    peerConnectionManager.ReceivedAbsolutePosition += receivedAbsolutePosition =>
    {
        Debug.Log($"[KHW] receivedAbsolutePosition: {receivedAbsolutePosition}");
    };
}
```

5. 결론

5.1. 기대효과

AR 기반 실내외 통합 길찾기 서비스의 기대 효과는 다음과 같다.

1. 길찾기 서비스를 실내외로 통합하고 실내의 특정 지점을 도착지로 설정할 수 있어 외부에서 건물로 들어간 후 목적지가 건물에 어느 곳에 있는지 찾는 시간을 단축할 수 있다.
2. 현실 세계와 가상 세계의 융합으로 사용자에게 새로운 환경을 제공하며 흥미를 돋우고 실제 환경을 기반으로 경로를 생성하므로 보다 정확하게 서비스를 제공할 수 있다.
3. 상대방이 시스템 내의 알고리즘으로부터 구현한 경로보다 더 빠른 경로를 알고 있다면 WebRTC를 통한 서버를 거치지 않은 P2P 실시간 통신으로 우수한 사용자 경험을 제공할 수 있다.

5.2. 추후 연구 방향

실내 공간을 안내할 때 현재 건물 정보를 미리 스캔하여 저장해야 한다는 번거로움이 있지만, 이 서비스가 상용화된다면 여러 사용자들이 스캔한 실내 정보를 클라우드에 저장하도록 하여 여러 건물의 실내 정보를 수집할 수 있다. 그리고 현재 1:1 통신만 지원하여 친구를 통한 길 안내 서비스는 오직 한사람이 다른 한사람에게만 가능하도록 설계되어있다. 추후 1:N 및 M:N 연결을 구축하도록 설계하면 더욱 효율적인 길 안내 서비스를 제공할 수 있다. 또한 WebRTC를 통한 다중 연결을

구축하면 서버를 거치지않고 통신할 수 있으므로 실시간으로 친구의 위치 정보를 파악할 수 있는
방향 등의 부가적인 서비스로 제공할 수 있다.

6. 참고문헌

- Anchor Reference: [https://developers.google.com/ar/develop/cloud-anchors?hl=ko%5D\(https%3A%2F%2Fdevelopers.google.com%2Far%2Fdevelop%2Fcloud-anchors%3Fhl%3Dko\)](https://developers.google.com/ar/develop/cloud-anchors?hl=ko%5D(https%3A%2F%2Fdevelopers.google.com%2Far%2Fdevelop%2Fcloud-anchors%3Fhl%3Dko))
- Polycam: <https://library.vuforia.com/using-3d-scans/model-targets-polycam>
- Outdoor Navigation: <https://assetstore.unity.com/packages/tools/integration/ar-gps-location-134882#reviews>
- ARway, Indoor Navigation: <https://www.youtube.com/watch?v=zq0wDY1jS7U>
- TURN Server: <https://webrtc.org/getting-started/turn-server?hl=ko>