

종합설계 프로젝트

최종결과보고서

프로젝트명	KPU AR Indoor navigation
팀번호	S1-9
문서제목	수행계획서(O) 2차발표 중간보고서(O) 3차발표 중간보고서(O) 4차발표 중간보고서(O) 최종결과보고서(O)

2020.11.20

팀원 : 강 필 수 (팀장)
김 태 훈
황 상 원

지도교수 : 이 정 준 교수 (인)

문서 수정 내역

작성일	대표작성자	버전(Revision)	수정내용	
2020.01.21	강필수(팀장)	1.0	수행계획서	최초작성
2020.03.02	강필수(팀장)	2.0	2차발표자료	설계서추가
2020.04.26	강필수(팀장)	3.0	3차발표자료	시험결과추가
2020.06.26.	강필수(팀장)	4.0	4차발표자료	시험결과 수정
2020.11.20	강필수(팀장)	5.0	최종결과보고서	최종작성

문서 구성

진행단계	프로젝트 계획서 발표	중간발표1 (2월)	중간발표2 (4월)	학기말발표 (6월)	최종발표 (10월)
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I. 서론 (1~6)	I II III
	II. 본론 (1~3)	II. 본론 (1~4)	II. 본론 (1~5)	II. 본론 (1~7)	
	참고자료	참고자료	참고자료	참고자료	

이 문서는 한국산업기술대학교 컴퓨터공학부의
 “종합설계” 교과목에서 프로젝트 “ARCore을 이용한 실내 비주얼
 내비게이션 시스템 설계 및 구현”을
 수행하는 강필수, 김태훈, 황상원 등이 작성한 것으로 사용하기
 위해서는 팀원들의 허락이 필요합니다.

목 차

I. 서론

1. 작품선정 배경 및 필요성
2. 기존 연구/기술동향 분석
3. 개발 목표
4. 팀 역할 분담
5. 개발 일정
6. 개발 환경

II. 본론

1. 개발 내용
2. 문제 및 해결방안
3. 시험시나리오
4. 상세 설계
5. Prototype 구현
6. 시험/ 테스트 결과
7. Coding & DEMO

III. 결론

1. 연구 결과

참고자료

I . 서론

1. 작품선정 배경 및 필요성

1. 작품선정 배경

- AR기술을 사용한 콘텐츠 제작
- 박물관이나 전시회와 같은 대규모 실내 공간에서 종종 길을 헤매는 경우가 발생한다.
- 실내에서는 GPS로 목적지의 정확한 위치를 파악하는 것이 어렵다.
- 실내에서 사용할 수 있는 내비게이션이 부족하다.

2. 필요성

- 내부가 복잡한 건물은 층마다 안내도가 있지만, 활용하기 힘들다.
- 평소에 안 가던 공학관에서 강의실 찾기가 힘들다.

2. 기존 연구 /기술동향 분석

- VIRNECT 국립중앙박물관 AR 도슨트 시범 서비스
스마트 글라스를 이용하여 박물관의 전시물에 AR 애니메이션이나 설명을 서비스한다.
- NaverLabs 인도어 AR 내비게이션
스마트폰 카메라로 받아들이는 이미지를 분석해 현재 위치를 파악한다.

3. 개발 목표

- 현재 위치에서 목적지까지의 경로를 AR로 표현한다.
- AR 기술을 이용해서 새로운 콘텐츠를 제작한다.
- 공학관 E동에 대한 Navigation을 제작한다.
- 특정 물체에 대한 안내를 AR 오브젝트로 구현한다.

4. 팀 역할 분담

강필수	사전 정의된 지도 제작 및 수정, positioning
김태훈	지도 내 특징점 학습, 애플리케이션 UI 제작
황상원	내비게이션 기능 및 경로에 따른 증강 기능 구현

5. 개발 일정

	12월	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월
계획 수립												
요구사항 수집												
분석 및 설계												
프로토타입 구현												
프로토타입 테스트												
프로그램 구현												
프로그램 테스트												
최종 마무리 작업												
논문 작성												

6. 개발 환경

운영체제 : Windows 10

IDE : Unity

언어 : C#

SDK : ARCore

실행환경 : Android 7.0이상

I . 본론

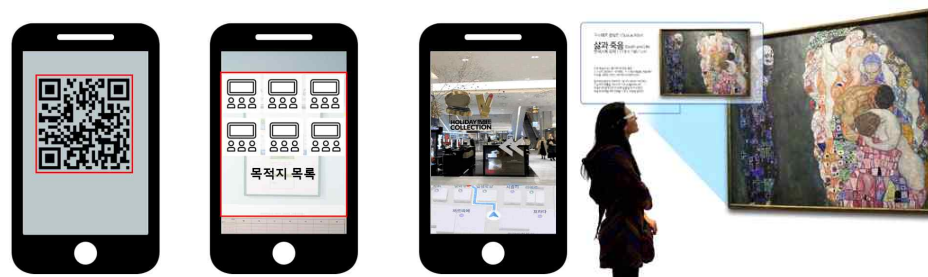
1. 개발 내용

1. 학교 시설 안전팀에서 받은 공학관 E동 건물의 평면도를 사용하여 Unity 상의 지도를 만들어 사람이 이동 가능한 구역과 불가능한 구역을 나누고 미니맵을 구현한다.
2. 카메라로 받아들이는 영상을 프레임 단위로 사용자의 움직임을 측정하여 Unity 지도상 사용자의 위치와 방향을 바꿔주는 Motion tracking & Rotation 기능을 구현한다.
3. 특정 목적지를 향해 경로 탐색 중에 다른 목적지를 선택할 시 경로를 재탐색할 수 있게 구현한다.

2. 문제 및 해결방안

1. 실시간으로 자신의 위치를 확인하는 positioning 기능의 부재로 고정된 한가지 위치에서만 가능하다. 이에 대한 해결방안으로 여러 장소에 QR 코드를 부착하여 여러 장소에서 사용할 수 있도록 개발한다.
2. 3D에서 두 회전축이 겹쳐져서 제대로 회전하지 않는 짐벌락 현상이 발생한다. 그래서 임의 축을 기준으로 회전 연산을 한 번에 처리하여 해결할 수 있게 도와주는 Quaternion클래스를 사용한다.

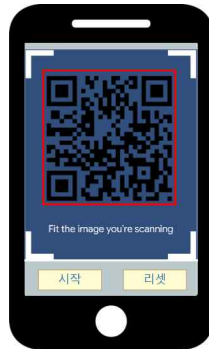
3. 시험시나리오



사전에 설정해둔 시작 포인트의 QR코드를 스캔하여 현재 자신의 위치를 동기화시킨다. 그 후 사용자는 AR 콘텐츠를 보며 이동이 가능해진다. 특정 물체를 터치할 경우 해당 물체에 대한 설명이 적힌 오브젝트를 띄운다. 특정 목적지를 선택할 시 해당 목적지까지의 경로를 화면의 미니맵에 표시하고 이동 방향을 AR 오브젝트로 표시해준다.

4. 상세 설계

❖ UI



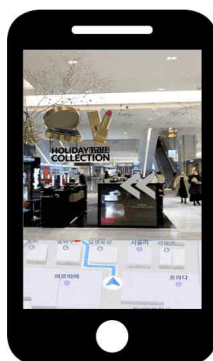
- QR Code 스캔화면

사용자의 현재 위치를 동기화하기 위해 QR코드를 이용한 스캔 기능을 사용한다. QR Code를 스캔할 사각형 영역을 구성하고 올바른 QR Code가 아닐 시 경고창을 띄운다. 만약 스캔에 성공했을 경우 흰색 안내문의 문구가 바뀌고 시작 버튼을 누를 수 있다. 만약 다시 스캔하고 싶을 경우 리셋 버튼을 눌러 스캔을 다시 할 수 있다.



- 목적지 설정 화면

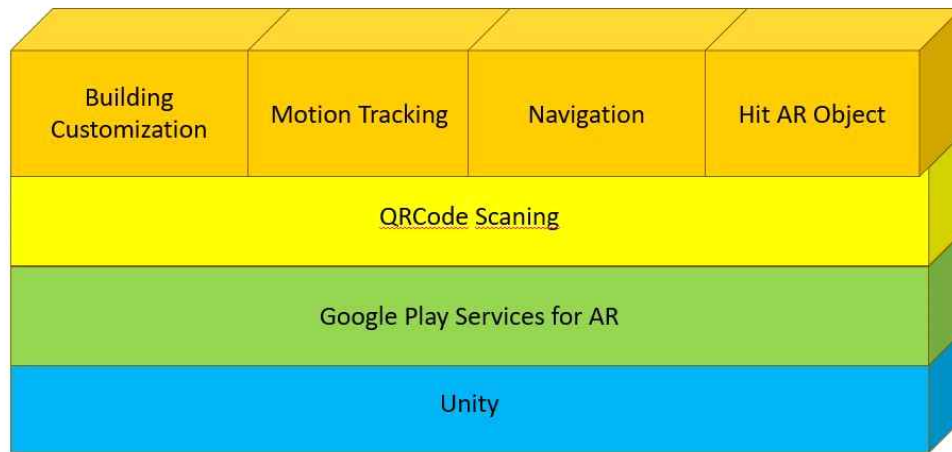
사용자가 현재 위치를 QR Code로 설정하면 목적지를 설정할 수 있다. 특정 목적지를 선택할 수 있게 드롭박스 및 팝업창으로 선택 가능한 장소들을 보여준다.



- 길 안내 화면

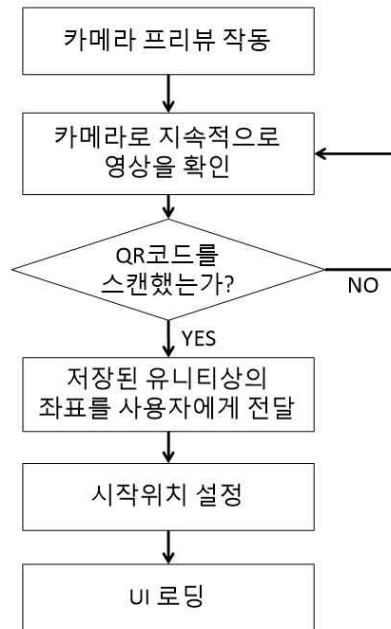
사용자가 목적지를 선택하게 되면 좌측 하단에 있는 미니맵에 경로를 표시함과 동시에 증강 현실에서 AR 콘텐츠를 경험할 수 있다.

❖ 아키텍처 구조



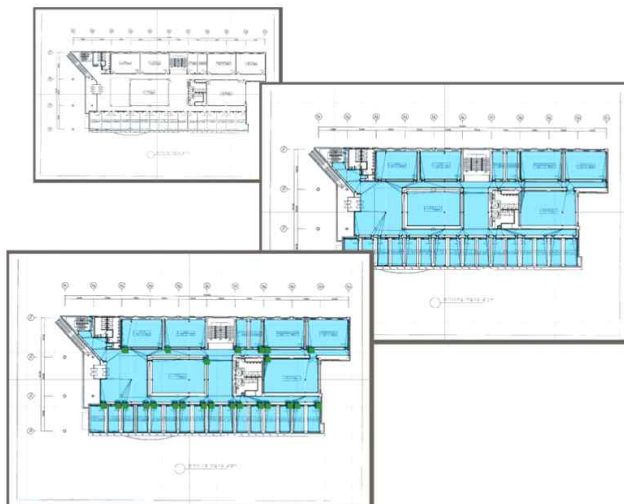
본 애플리케이션이 AR기술을 사용하지만, Unity를 사용하여 만들어진다. AR Core는 Google Play Service for AR와 같은 AR 지원 애플리케이션이 있어야 사용할 수 있다. 애플리케이션은 사용자의 위치를 QR Code로 확인하는 QR Code Scanning 기능, 애플리케이션을 사용할 건물의 내부 지도를 구성하는 Building Customization, 만든 지도의 특정 목적지를 선택할 수 있는 Set Destination, 사용자의 움직임에 따라 위치를 변경시키는 Motion Tracking, 카메라가 회전할 때 Unity의 객체도 회전시키는 Rotation, 사용자 위치에서 선택한 목적지까지 경로를 탐색하는 Navigation, 터치해서 AR Object와 상호작용하는 Hit AR Object 기능으로 이루어져 있다.

❖ QR Code Scanning



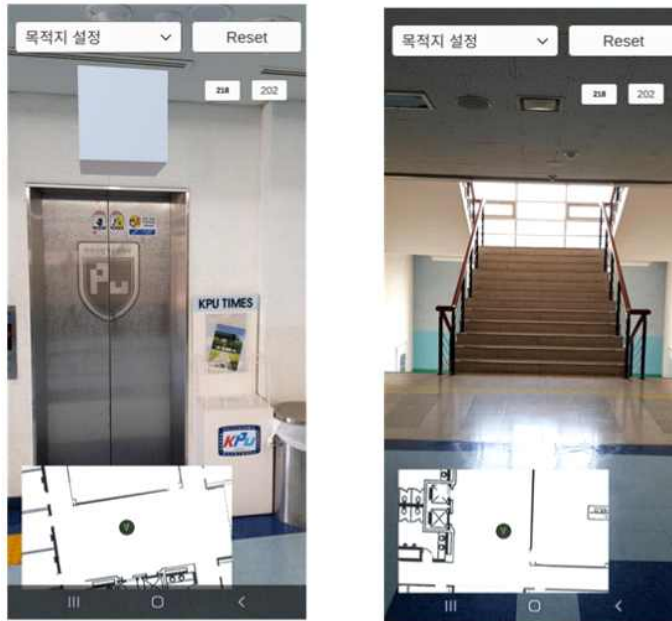
- 사용자의 현재 위치와 바라보는 방향을 동기화하기 위해 구글에서 배포하는 QR코드 스캔 라이브러리인 Zxing을 이용하여 스캔하면 저장된 Unity 상의 좌표를 사용자에게 전달해서 시작 위치를 설정한다.

❖ Building Customization



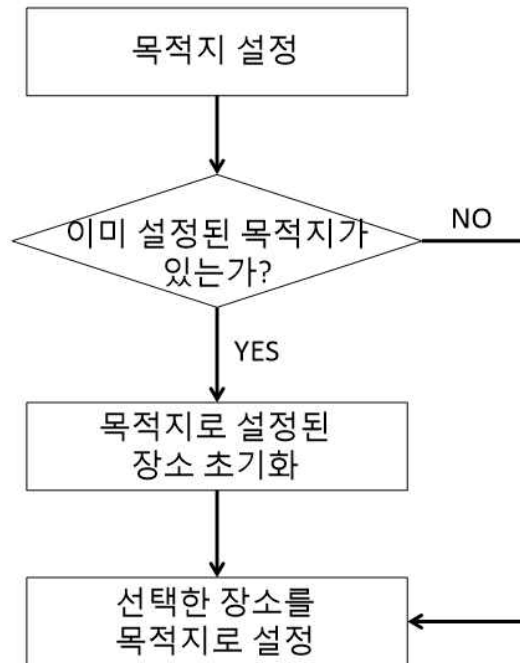
- 공학관 평면도를 기반으로 이동 가능한 구역과 이동 불가능한 구역을 설정한 후 목적지가 될 장소들을 설정한다.

❖ Motion Tracking & Rotation



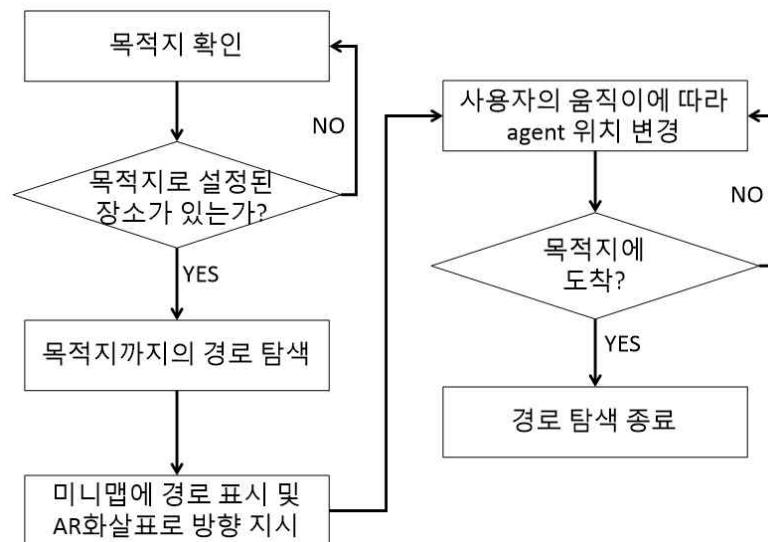
- ARCore의 Frame클래스를 사용하여 앱 실행 시 카메라가 비추는 곳의 프레임과 움직이는 주체인 agent의 방향을 동기화시킨다.
- 사용자가 움직이면 이전 위치와 움직인 현재 위치의 변화 값을 Unity 상의 agent에게 전달하여 이동시킨다.
- 프레임마다 업데이트되는 AR카메라의 현재 위치와 이전 위치에 차이를 구한 값만큼 agent의 위치를 이동시킨다.
- AR카메라의 움직임에 따라 Unity 상의 agent의 Y축을 회전시킨다.
- 3차원에서 물체를 회전시킬 경우 회전시키는 축의 순서에 따라 바라보는 방향이 달라지는 짐벌락 현상이 발생한다.
- 이를 방지하기 위해 Quaternion클래스와 보간법을 사용한다.

❖ Set Destination



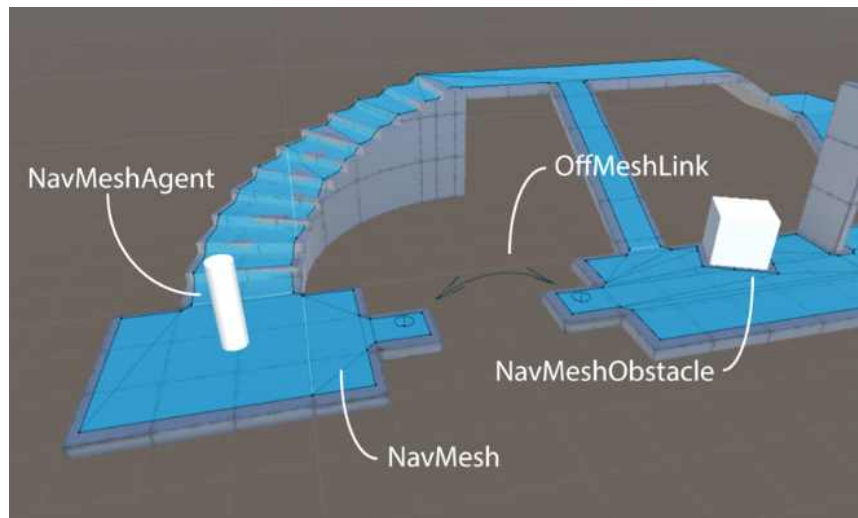
- 이미 목적지가 설정된 상태일 경우 해당 강의실을 초기화시키고 선택한 강의실을 목적지로 설정한다.

❖ Navigation



- 목적지로 선택된 장소가 있으면 목적지까지의 경로를 Unity의 Navmesh기능을 이용하여 탐색
- 미니맵에 경로를 표시하고 AR상에서 화살표로 방향을 가리킨다. 사용자 움직임과 위치에 따라 경로를 재탐색

❖ Navmesh



- Unity에 내제된 내비게이션 기능으로써 간단한 길찾기 시스템을 만들 수 있다. 기본 시스템은 아래와 같이 구성되어 있다.

- Navmesh : 네비게이션 메쉬의 줄임말로, 한 지형에서 걸을 수 있는 표면을 뜻하며, 내비메쉬를 사용하여 이동할 수 있는 경로를 찾을 수 있다. 데이터 구조는 레벨 지오메트리에서 자동으로 빌드 또는 베이킹 된다.

- Navmesh Agent : 네비메쉬 에이전트는 한 지형에서 움직이는 객체, 즉 캐릭터를 의미한다. 에이전트는 앞서 설명한 네비메쉬상에서 움직일 수 있고 여러 개의 에이전트가 있으면 서로를 피할 수 있다.

- 오프 메시 링크(Off-Mesh Link)

오프 메시 링크를 사용하여 걸을 수 있는 표면만으로는 정의할 수 없는 내비게이션 단축기를 통합할 수 있다. 예를 들어 배수로나 울타리를 뛰어넘거나, 문을 지나가기 전에 여는 행동등을 모두 오프 메시 링크로 정의할 수 있다. 그러나 본 프로젝트에서는 사용되지 않는다.

- 내비메시 장애물(NavMesh Obstacle)

컴포넌트를 사용하여 에이전트가 한 지형을 탐색하는 동안 회피해야하는 움직이는 장애물을 정의할 수 있다. 물리 시스템이 제어하는 통이나 상자를 예를 들 수 있다. 움직이는 장애물이라면 에이전트가 이를 피하도록 하고, 장애물이 정지한 경우 내비메시에 구멍을 카빙하여 에이전트가 장애물을 돌아가도록 경로를 변경하거나, 정지한 장애물이 경로를 완전히 차단할 경우 다른 경로를 찾게할 수 있다. 하지만 본 프로젝트에서는 사용되지 않는다.

네비메쉬는 다음과 같은 특징을 갖는다

- 이동과 길찾기를 위해서 3D공간을 2D형식으로 변환
- 장애물을 피해서 움직일 수 있는 자유로운 공간을 설정
- Linked List 형태의 자료구조로 구현

❖ Navmesh

네비메시를 활용한 길찾기 알고리즘의 작동 프로세스는 다음과 같다.

- 1) 출발점과 도착점을 연결하는 벡터를 초기 이동 경로로 설정
- 2) 네비메시에 걸쳐있는 삼각형들과 이동 경로 벡터를 비교
- 3) 삼각형으로부터 벗어난 이동 경로가 있다면 인접한 삼각형으로 이동 경로를 변경
- 4) 이동 경로 벡터가 삼각형 안에서만 움직일때까지 2~3을 반복.

❖ A*(A Star) Algorithm

유니티의 네비메쉬에서 길찾기 기능에 사용되는 알고리즘이다. 보통 길찾기 알고리즘이라고 하면 대부분 다익스트라 알고리즘을 생각하는데 실제 프로그램이 적용할 때 문제가 되는 부분이 많다. 다익스트라 알고리즘은 경로를 찾을 때 각 노드에서 목적지까지의 모든 경로에 대한 실제 비용을 계산하여 비교하기 때문에 알고리즘 수행 시간이 많이 든다. 이러한 문제를 해결하기 위해 다음 노드로의 분기를 모든 노드가 아닌 가장 적은 추정 비용을 갖는 노드로 이동할 수 있게 휴리스틱 함수(Heuristic Function)를 적용하면 각 노드에서 분기를 줄일 수 있다. A*알고리즘이 대표적인 휴리스틱 함수를 사용하는 알고리즘이다.

A*알고리즘은 추정 비용을 계산해 가장 적은 비용을 갖는 노드를 선택해 탐색하며 다음 노드로의 분기를 줄일 수 있다. 목적지까지의 예상 비용만을 비교하여 다음 노드를 선택하는 BFS와 다른 점은, 노드를 선택하는 기준으로 시작점으로부터 노드까지 이동하는데 드는 비용과 현재 노드로부터 목적지까지 예상되는 비용을 더한 값을 사용한다.

BSF의 선택기준 = 목적지까지 예상비용

A*의 선택기준= 추정비용= 시작점으로부터의 비용 + 목적지까지의 예상비용

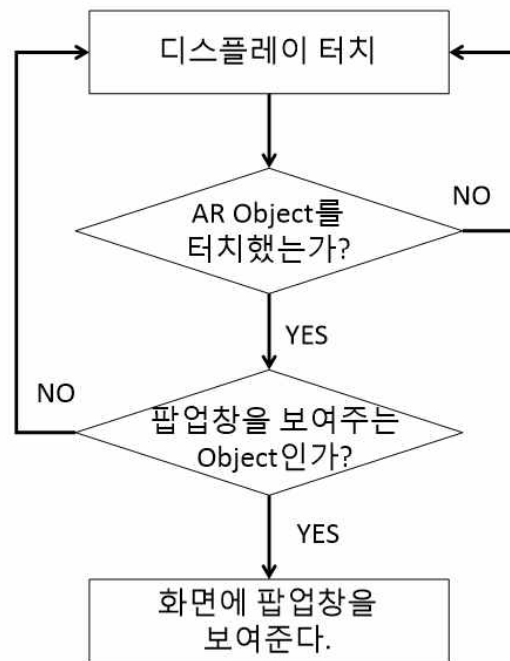
A*는 다음으로 탐색할 노드를 선택할 때 오픈 리스트와 클로즈 리스트를 사용한다. 오픈 리스트는 탐색이 아직 안된 추정치만 존재하는 노드들의 집합을, 클로즈 리스트는 탐색이 되어 추정비용이 계산된 비용으로 바뀐 노드들의 집합을 의미한다. 현재 노드에서 클로즈 리스트에 들어있지 않는 인접한 이웃 노드들의 추정치를 계산하고 각 이웃 노드의 부모 노드를 현재 노드로 지정한다. 다음으로 이웃 노드들을 오픈 리스트에 넣는데, 이미 오픈 리스트에 해당 노드가 존재할 경우 추정 비용을 비교해 작은 값을 갖는 노드를 사용한다. 그 후, 오픈 리스트에 저장된 노드 중 가장 작은 추정 비용을 갖는 노드를 다음으로 탐색할 노드로 선택한다. 선택된 노드는 탐색이 된 노드이기 때문에 오픈 리스트에서 클로즈 리스트로 옮기고 다시 현재 노드에 인접한 노드들의 추정 비용을 계산해 오픈 리스트에 넣는다. 목적지 노드가 이웃 노드로 나타날 때 까지 위의 과정을 반복하는 것이 A*알고리즘의 작동 방식이다.

A*알고리즘은 시작점으로부터의 비용을 추가적으로 사용한다. 그리고 각각의 요소에 가중치를 적용한다. 따라서 다음과 같은 식으로 표현할 수 있다.

A*의 선택기준= 추정비용= a*시작점으로부터의 비용 + b*목적지까지 예상 비용

일반적으로 말하는 A*알고리즘의 경우 a와 b의 값을 1:1의 비율을 사용한다. 이 값들을 바꿈으로서 같은 알고리즘으로 다른 결과를 얻을 수 있다.

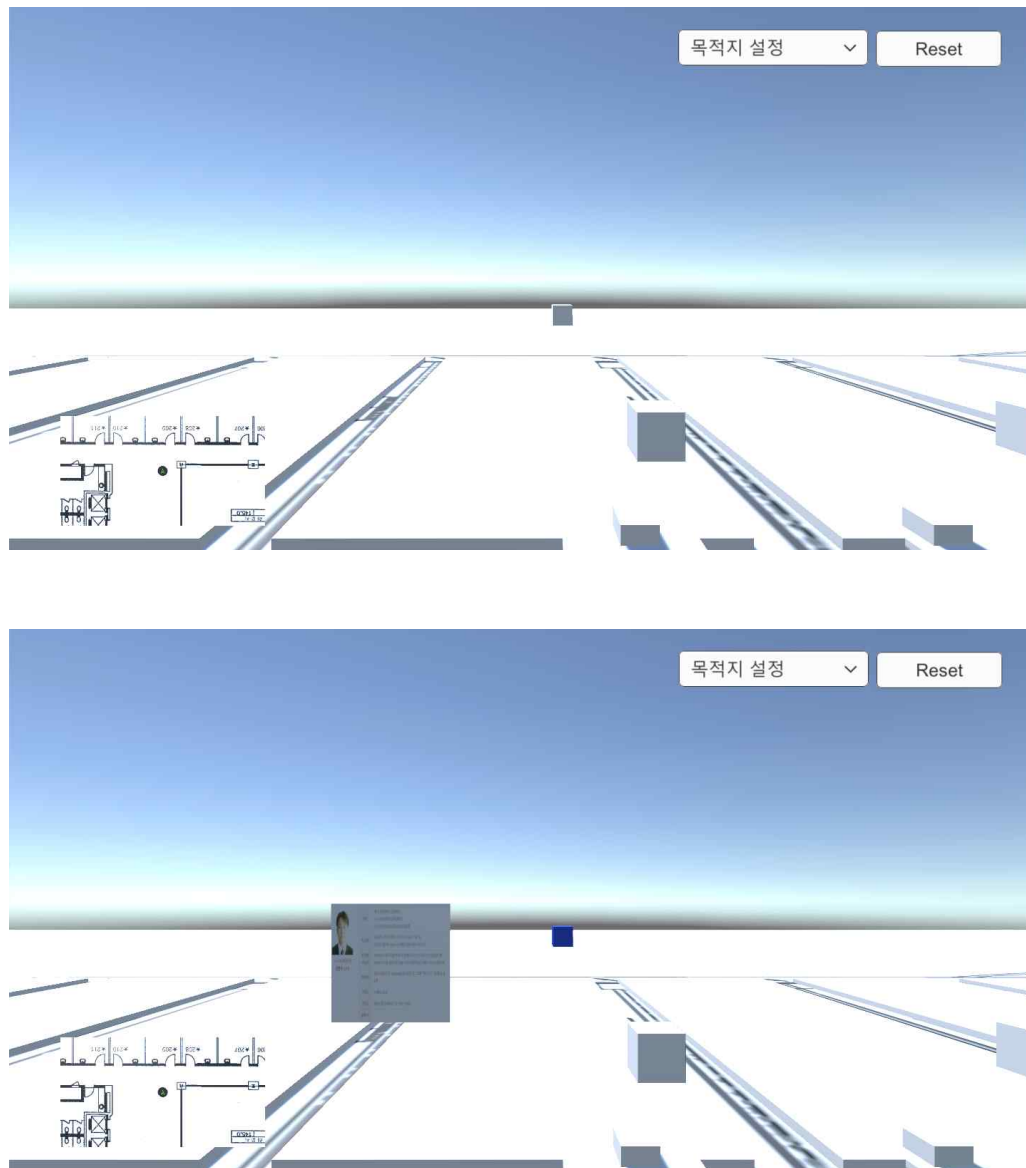
❖ Hit AR Object



- 특정 위치의 ARobject를 터치할 시 그에 대한 설명이 적혀있는 팝업창을 열어 사용자에게 보여준다.

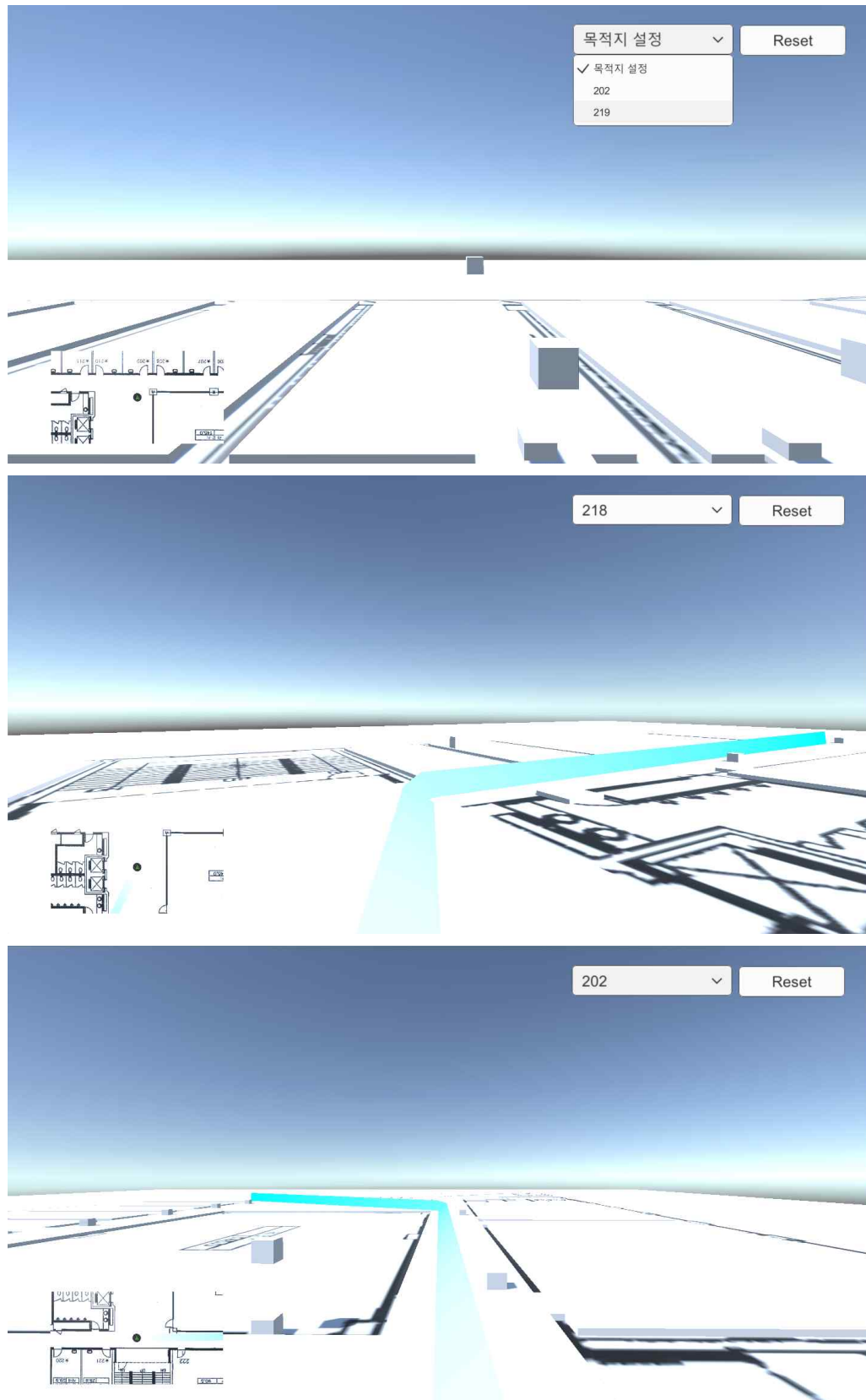
5. Prototype 구현

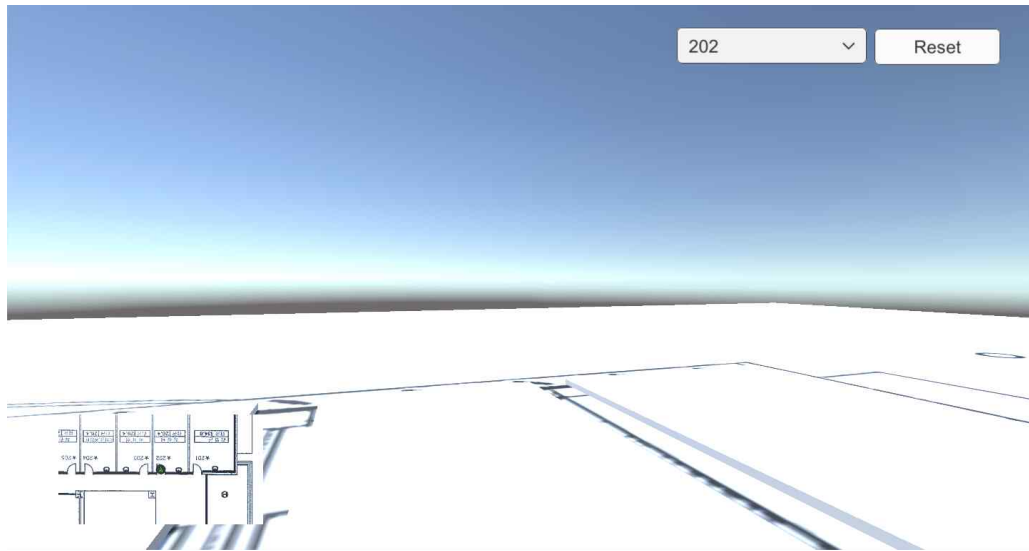
❖ QR Code Scanning



특정 AR오브젝트를 터치할 경우 해당 오브젝트와 연결되어있는 관련 정보창을 표시한다.

❖ Navigation





사용자가 움직이거나 방향을 바꾸었을 때 위치나 방향이 일치하며 목적지를 설정할 경우 해당 목적지까지 경로를 표시해주며 도중에 목적지를 바꿔도 바뀐 목적지로 경로를 문제없이 표시한다.

6. 시험 / 테스트 결과

체크리스트	
카메라로 QR코드를 제대로 인식하는가?	○
QR코드 읽은 정보를 제대로 저장해서 Agent에게 넘겨주는가?	○
QR코드로 읽은 출발지의 위치를 제대로 인식하는가?	○
QR코드를 인식한 뒤 AR화면으로 넘어가는가?	○
평면도에서 움직일 수 있는 영역을 제대로 지정했는가?	○
실제 움직이는 위치와 Unity상의 변경된 위치가 일치하는가?	△
AR기기를 회전했을 때 바라보는 방향이 일치하는가?	○
짐벌락 현상이 안 일어나도록 하였는가?	○
AR오브젝트 정해진 위치에 제대로 증강되어 있는가?	○
화면을 터치했을 때 터치한 AR오브젝트를 제대로 인식하는가?	○
AR오브젝트를 터치했을 때 연결된 정보창이 나오는가?	○
다시 AR오브젝트를 터치했을 때 정보창은 사라지는가?	○
AR오브젝트가 현실과 위화감이 없는가?	△
미니맵은 사용자가 움직였을 때 사용자를 제대로 따라오는가?	○
목적지를 선택할 경우 경로를 표시해주는가?	○
표시한 경로는 목적지를 제대로 가리키고 있는가?	○
Reset버튼을 눌렀을 때 경로는 제대로 사라지는가?	○
사용자와 목적지 사이를 제대로 잇고 있는가?	○

❖ Motion Tracking & Rotation

1. 사용자의 위치가 달라지는 문제 발생

- 특정 구간을 지나갈 때 미니맵상의 사용자의 위치가 잠깐 달라지는 현상이 발생한다.

-> 이 문제가 하드웨어적 문제인지 소프트웨어적 문제인지 확인 안 되는 상태이다.

❖ Augmented Reality

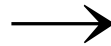
1. 벽 너머의 AR 오브젝트 노출

- Unity의 AR 카메라는 디바이스 카메라의 영상과 Unity에 있는 오브젝트 모두를 볼 수 있다. 그리고 이를 필터링해서 Unity에서 AR 오브젝트만 보이게 설정했기 때문에 실제로 벽 너머에 있는 보이지 않아야 하는 AR 오브젝트가 보이게 된다.

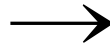
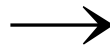
-> 사용자와의 거리를 계산해서 가까운 오브젝트만 보이게 설정한다.

7. Coding & DEMO

❖ QR Code Scanning



❖ Navigation



Ⅲ. 결론

1. 연구 결과

❖ 결론 및 향후 발전 방향

- 본 논문에서는 실내에서 사용 가능한 AR 내비게이션 시스템의 설계 및 구현에 대해 설명하였다. 이를 위하여 Google에서 서비스하는 ARCore와 QR코드를 이용하여 현재 위치를 파악할 수 있도록 구현하였다.
- 본 시스템은 대규모 박물관이나 전시회에서 원하는 목적지까지 쉽게 갈 수 있도록 안내하고 AR을 활용하여 관람객들에게 새로운 경험을 선사할 수 있고, 관심 없는 사람들에게도 흥미를 유발하게 시켜 관람객을 유치하는데 이바지할 수 있다.
- 향후 이러한 시스템을 홀로렌즈 등에 결합하면 핸드폰을 들고 다니지 않고도 사용자 편하게 관람할 수 있도록 발전할 수 있다.

❖ 최종 결과물





참고자료

1. 절대강자 유니티 VR/AR

위키북스

2. AR을이용한캠퍼스정보안드로이드앱

<https://fist0512.tistory.com/66>

3. Unity로AR의실내 탐색

<https://blog.mapbox.com/indoor-navigation-in-ar-with-unity-6078afe9d958>

4. Unity에서 ARCore 기반 실내 내비게이션 애플리케이션 만들기

<https://blog.raccoons.be/arcore-powered-indoor-navigation-unity>

5. GPS가 없는 실내에서의 지도 서비스는 어떻게 할까?

<https://medium.com/naver-cloud-platform/gps%EA%B0%80-%EC%97%86%EB%8A%94-%EC%8B%A4%EB%82%B4%EC%97%90%EC%84%9C%EC%9D%98-%EC%A7%80%EB%8F%84-%EC%84%9C%EB%B9%84%EC%8A%A4%EB%8A%94-%EC%96%B4%EB%96%BB%EA%B2%8C-%ED%95%A0%EA%B9%8C-3763da033e72>

6. [Beacon] 1. 비콘에 대해 알아보기 - 공대오빠의 인생살이

<https://gongdae58.tistory.com/72>

7. ARCore Develop

<https://developers.google.com/ar/develop>