

22 팀

최종 보고서

Graduation Project

3194 Class

Date

2021-11-26

Team 22 Information

201812663 송나은

목차

1. 프로젝트 소개

- 1.1. 구성원 소개
- 1.2. 주제 선정 배경
- 1.3. 목표

2. 개발 환경

- 2.1. 개발 환경
- 2.2. 기타 모듈

3. 요구사항 분석

- 3.1. 사용자 다이어그램
- 3.2. 기능 요구사항
- 3.3. 비기능 요구사항

4. 핵심문제 정의 및 해결방안

- 4.1. 핵심문제 정의
- 4.2. 관련 연구 개발 정리 및 참고사항
- 4.3. 본 팀의 해결 방법

5. 디자인

- 5.1. Deployment Diagram
- 5.2. Class Diagram
- 5.3. Sequence Diagram

6. 결과물

- 6.1. 회원가입과 로그인
- 6.2. 마커 인식
- 6.3. 메모 열람과 메모 작성

7. Risk Analysis + Risk Reduction Plan

8. Success Criteria

8.1. 테스트 방법

8.2. 테스트 결과

8.3. 테스트 결과 분석

9. 개선사항

10. 참고자료

1. 프로젝트 소개

졸업 작품명은 '마커 인식을 이용한 가상메모 AR 애플리케이션'이다.

1.1. 구성원 소개

아래 첨부된 표에서는 22팀 나모지의 구성원을 소개한다.

순번	이름	학번	역할	전화번호	이메일
1(대표)	송나은	201812663	대표, 총괄	010-6685-2912	ene607@naver.com

1.2. 주제 선정 배경

평소 AR 기술에 관심이 많았다. 명품 브랜드에서 AR 애플리케이션으로 사용자에게 신발 피팅을 제공한다는 뉴스를 보며, AR 콘텐츠는 점점 수요가 높아질 거라 예상했기 때문이다. 그래서 졸업 프로젝트에서 AR 기술을 활용하고자 했다.

AR은 사용자의 기기를 통해서만 가상의 객체를 볼 수 있다. 즉, AR이 프라이버시를 보호하는 수단이 될 수 있음을 의미한다. 프라이버시가 보호되어야 하는 상황에서 AR을 어떻게 활용하면 좋을지 고민했다. 그러던 중, AR을 통해 내가 원하는 장소에 다른 사람에게 보여주고 싶지 않은 메모를 남길 수 있는 애플리케이션을 구상하게 되었다.

1.3. 목표

마커를 인식하여 가상객체를 배치하는 기술은 ARToolKit과 같은 이전의 기술에서도 사용하였을 만큼 널리 사용되는 기술이다.

본 애플리케이션은 사용자의 모바일 폰으로 현실세계에서 쉽게 접할 수 있는 이미지를 마커로 사용함으로써 모바일 폰 속 증강현실에서의 몰입감을 깨뜨리지 않는 것을 목표로 한다. 또한, 마커가 인식된 위치에 가상메모라는 가상객체를 제공함으로써 현실세계와 가상공간의 협응이 가능해지도록 한다. 제작된 애플리케이션은 다수의 가상메모를 동시에 스폰(Spawn)한다. 본 프로젝트에서는 가상메모 스폰 위치의 정확도를 높이는 방법을 고안하고 제안하는 것을 목표로 한다.

2. 개발 환경

2.1. 개발 환경

① 활용한 SW: 상용 SW인 SQLite Database를 사용.

- ② Platform Layer: Android studio, Unreal Engine 4.26.2, ARCore
- ③ Infra Layer
 - OS: Android
 - Machine: Android mobile phone(Samsung Galaxy S7 edge(SM-G935))
- ④ Framework: ARCore
- ⑤ API: Android API 30.0.3, ARCore API
- ⑥ DBMS: USQLite Database

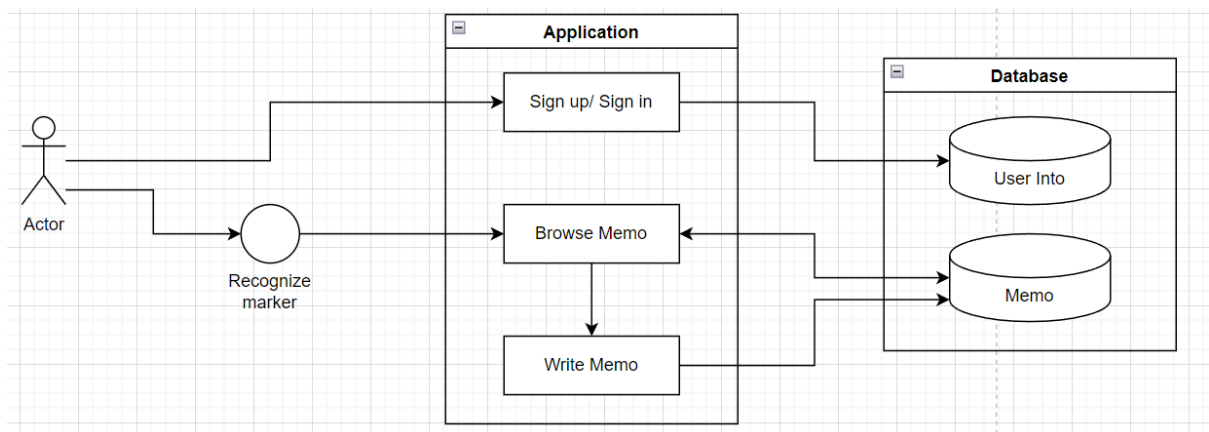
2.2. 기타 모듈

- ① ARCore SDK for Unreal v1.7.0
- ② Android NDK 21.1.6352462

3. 요구사항 분석

3. 절 요구사항 분석에서는 중간 보고서 내용에서 수정된 내용이 포함되어 있다.

3.1. 사용자 다이어그램



3.2. 기능 요구사항

- ① Sign up/ Sign in

유스케이스 명	Sign up/Sign in
액터	주 액터: 사용자

목표	<ul style="list-style-type: none"> - 사용자는 입력한 정보를 바탕으로 자신의 계정을 만들 수 있다. - 사용자는 계정을 입력하여 애플리케이션에 접속할 수 있다.
시작 조건	회원가입 과정이다. 사용자는 사용하고자 하는 계정을 입력하여 계정을 생성한다. 회원가입에 성공하면 새로운 ID와 비밀번호가 회원 정보 데이터테이블에 추가된다.
이후 조건	로그인 과정이다. 사용자가 회원 정보 데이터테이블에 존재하는 ID와 그 ID에 해당하는 비밀번호를 입력해야 로그인에 성공한다.
정상적인 흐름	<p>어플리케이션의 첫 접속인가에 따라 두 가지 흐름이 존재한다.</p> <ul style="list-style-type: none"> - 어플리케이션에 처음 접속하는 경우 <ol style="list-style-type: none"> 1. 사용자는 회원가입 위젯에서 사용하고자 하는 계정을 입력 후, Confirm 버튼을 누른다. 2. 로그인 위젯으로 이동하며, 사용자는 자신의 계정을 입력한다. 3. 회원 정보 데이터테이블에 존재하는 계정인 경우에는 로그인에 성공하고, 존재하지 않는 계정인 경우에는 로그인에 실패한다. - 어플리케이션에 처음 접속하는 것이 아닌 경우 <ol style="list-style-type: none"> 1. 로그인 위젯에서 사용자는 자신의 계정을 입력한다. 2. 회원 정보 데이터테이블에 존재하는 계정인 경우에는 로그인에 성공하고, 존재하지 않는 계정인 경우에는 로그인에 실패한다.

② Recognize marker

유스케이스 명	Recognize marker
액터	<p>주 액터: 사용자</p> <p>부 액터: 모바일 폰</p>
목표	모바일 폰의 후면 카메라로 현실세계의 마커를 인식하여 현실세계의 정보를 증강현실에서 활용한다.
시작 조건	사용자는 모바일 폰의 후면 카메라로 애플리케이션에서 미리 지정한 마커를 인식한다. 마커가 인식된 상태에서는 사용자의 모바

	일 폰 왼쪽 상단에 '마커의 종류: Tracking'이라는 문구가 뜬다.
이후 조건	한 번 인식하여 트래킹 하던 마커를 더 이상 인식하지 않으면 사용자의 모바일 폰 왼쪽 상단에 '마커의 종류: Not Tracking'이라는 문구가 뜬다.
정상적인 흐름	<ol style="list-style-type: none"> 1. 사용자가 마커를 인식하면 해당 마커에 대해 Tracking 상태가 되며 가상메모가 등장하고 이후의 사용을 진행할 수 있다. 2. 더 이상 마커를 인식하지 않으면 Not Tracking 상태가 되며 이후의 과정을 더 이상 진행할 수 없다.

③ Browse memo

유스케이스 명	Browse memo
액터	주 액터: 사용자
목표	사용자가 마커를 인식하면, 메모 정보 데이터테이블에서 해당 마커에 대한 메모 내용을 가져와서 사용자에게 제공한다.
시작 조건	사용자가 마커를 인식하면, 메모 정보 테이블에서 해당 마커의 메모 내용을 가져오려고 시도한다.
이후 조건	메모 정보 테이블에 저장된 메모 내용이 있을 경우, 사용자 모바일 폰 디스플레이에 스폰된 가상메모에 읽어온 메모 내용이 주어진다. 저장된 메모 내용이 없을 경우, 디스플레이에는 빈 메모 내용을 가진 가상메모가 스폰된다.
정상적인 흐름	<ol style="list-style-type: none"> 1. 마커를 인식 2. 메모 정보가 있는/없는 가상메모가 사용자 모바일 폰 디스플레이에 주어짐 3. 메모 내용을 작성/수정할 수 있는 Memo 버튼이 주어짐

④ Write memo

유스케이스 명	Write memo
액터	주 액터: 사용자
목표	사용자는 인식한 마커에 대한 메모 내용을 작성하고, 작성된 메모

	내용은 메모 정보 데이터테이블에 저장된다.
시작 조건	사용자가 마커를 인식하면, Memo 버튼이 주어진다. Memo 버튼을 누르면 인식된 마커를 통해 스폰되는 가상메모의 내용을 작성/수정할 수 있는 메모 작성 위젯이 주어진다.
이후 조건	메모 작성 위젯에 저장하고자 하는 메모 내용을 작성하고 Save 버튼을 누르면, 사용자가 작성한 메모 내용이 메모 정보 데이터테이블에 저장된다.
정상적인 흐름	1. Memo 버튼을 누름 2. 메모 작성 위젯에서 메모 내용을 작성하고 Save 버튼을 누름 3. 작성된 메모는 메모 정보 데이터테이블에서 갱신됨

3.3. 비기능 요구사항

- 제품 요구사항

① 사용성

사용자는 애플리케이션을 실행시키고 로그인해서 마커를 인식하는 과정만으로 가상메모를 열람하거나 작성할 수 있다.

② 효율성

- 마커가 인식되어 해당 마커에 저장된 메모를 사용자의 기기에 띄우기까지는 1초 내에 이루어져야 한다.

- 사용자가 작성한 메모는 1초 이내에 데이터베이스에 저장되어야 한다.

③ 신뢰성

사용자는 메모 인식, 메모 작성, 메모 열람을 실패할 가능성이 낮아야 한다.

④ 이식성

나모지는 안드로이드 모바일 폰 내에서 작동할 수 있다.

- 조직 요구사항

① 배포: SW는 .apk 파일의 형태로 배포된다.

② 구현: SW 구현을 위해서는 Unreal Engine 4.26.2와 AR Core를 사용한다. 추가 되는 구

현 언어는 C++이다.

- 제약사항

사용자는 개발 비용의 부족으로 사물이 아닌 마커를 인식해서 가상 메모를 작성하는데, 프로그램의 흐름을 고려할 때는 마커를 현실 세계의 사물로 생각한다.

4. 핵심문제 정의 및 해결방안

5. 절에서는 중간 발표 이후의 핵심문제를 다룬다.

4.1. 핵심문제 정의

중간 발표 이후에서의 핵심문제는 여러 개의 가상메모를 스폰하였을 때, 가상메모가 의도한 위치에 스폰되는 것이다. 여러 마커를 인식하여 동시에 가상메모를 스폰 시켜, 사용자에게 여러 메모 정보를 동시에 전달하고자 하였다. 해당 기능은 정보를 제공하는 기능이 추가 되는 팀의 애플리케이션에 적합하다고 생각하여 해당 기능의 구현에 비중을 두었다. 그러나 여러 마커에 대해 동시에 가상메모를 스폰하려고 할수록, 스폰 성공확률이 떨어지는 경향을 보였다.

4.2. 관련 연구 개발 정리 및 참고사항

① 마커 기반 가상객체 생성

Shelton and Hedley(2002)는 증강현실 개발 소프트웨어 ARToolKit을 사용하여 증강현실 지리학습 콘텐츠를 개발하였고, 그 효과를 실험했다. 증강현실 콘텐츠를 이용하여 학습한 학생들은 공간적 개념과 관련성에 대해 기존보다 높은 이해도를 보였으며, 학업 성취도도 증가했다.

이러한 기존 연구를 통해 FLARToolKit을 사용하여 마커를 인식하고 AR 지구의의를 제작하는 콘텐츠가 제작되었다. 해당 콘텐츠는 ArcGIS로 주제도를 제작한 후, FLARToolKit의 가상 구 표면에 입혔다. 마커를 웹캠으로 인식하여 가상 지구의의를 제공하였다. 시스템에서 필요한 마커는 FLARToolKit에서 기본적으로 제공하고 있는 마커를 사용하였다.

언급한 선행 연구는 마커 인식을 사용하였지만, 사용자의 몰입도를 고려하지 않은 마커를 사용하였다. 증강현실도 가상현실과 마찬가지로 사용자가 가상세계에 접속하므로, 가상세계에 대한 사용자의 몰입도가 깨지지 않는 것은 중요하다. 따라서 본 프로젝트에서는 일상생활에서 자주 마주할 수 있는 사물의 이미지를 마커로 사용하여, 사용자의 가

상세계에 몰입도를 고려하였다.

② 증강현실에서의 가상 공간좌표 형성

3차원 위치측정을 위해 비콘을 사용하였다. 비콘은 마커와 같이 증강현실 정보를 제공하는 역할과 3차원 좌표를 구하는 역할을 가진다. 비콘을 사용해서 3차원 좌표를 구하는 방법은 미리 위치를 알고 있는 비콘과의 각 거리 값을 3차원 삼각 측량법 계산을 사용하여 3차원 좌표 값을 측정하는 방법이다. 해당 연구에서 사용한 기법은 조명이 매우 밝은 상황에서도 증강현실에서의 가상 공간 좌표를 구현하는 것에 성공했다. 그러나 마커 기반에 비해 속도가 느리다는 단점이 존재했다.

본 프로젝트에서 제작하고자 하는 콘텐츠는 주 사용 공간이 현실세계의 일반적인 공간임을 고려하였을 때, 속도가 더 빠른 마커 기반이 더 적합하다고 판단하였다.

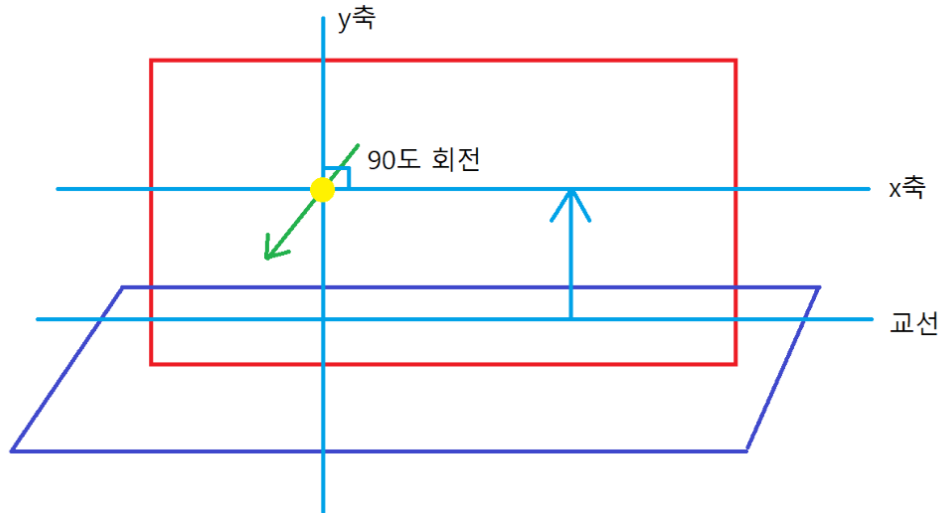
4.3. 본 팀의 해결 방법

관련 연구를 살펴보면, 마커를 사용할 때 사용자의 몰입도를 고려하여야 함을 알 수 있었다. 따라서 중간발표 이전에 일상생활에서 쉽게 마주할 수 있는 이미지를 마커로 사용함으로써 해당 문제를 극복하기로 하였다.

마커를 사용하는 것은, 비콘과 같은 다른 수단에 비해 비교적 빠른 속도를 보이기 때문에 마커의 사용은 적합하다. 그러나 여러 개의 마커를 인식하여 동시에 가상메모를 스폰하였을 때 정확도가 문제되었다. 해당 문제에 대한 분석은 9.3. 절을 참고한다.

9.3. 절에서는 인식된 마커들이 공유하는 하나의 좌표계가 만들어진다면, 각 마커의 로컬 좌표계 간섭으로 인한 문제가 개선될 것이라고 예상하였다.

본 팀은, 언급한 해결방안을 구현하기 위한 방법을 구상하였다. 3차원 위치 좌표계를 생성하는 방법은 아래의 그림을 참고한다.



우선, 모바일 폰을 통해 수직관계에 있는 두 개의 평면을 인식해서, 두 평면에 대한 평면의 방정식을 구한다. 인식된 평면의 수직 벡터와 평면 위의 한 점을 이용하여 평면의 방정식을 구한다. 평면의 방정식 P 를 구하는 공식은 아래와 같다.

$$P = n \cdot (x - x_0)$$

좌표계의 x 축을 구하기 위해, 두 평면 P_1 과 P_2 의 교선의 방정식을 구한다. 교선의 방정식을 구하는 공식은 아래와 같다.

$$P_1 = P_2$$

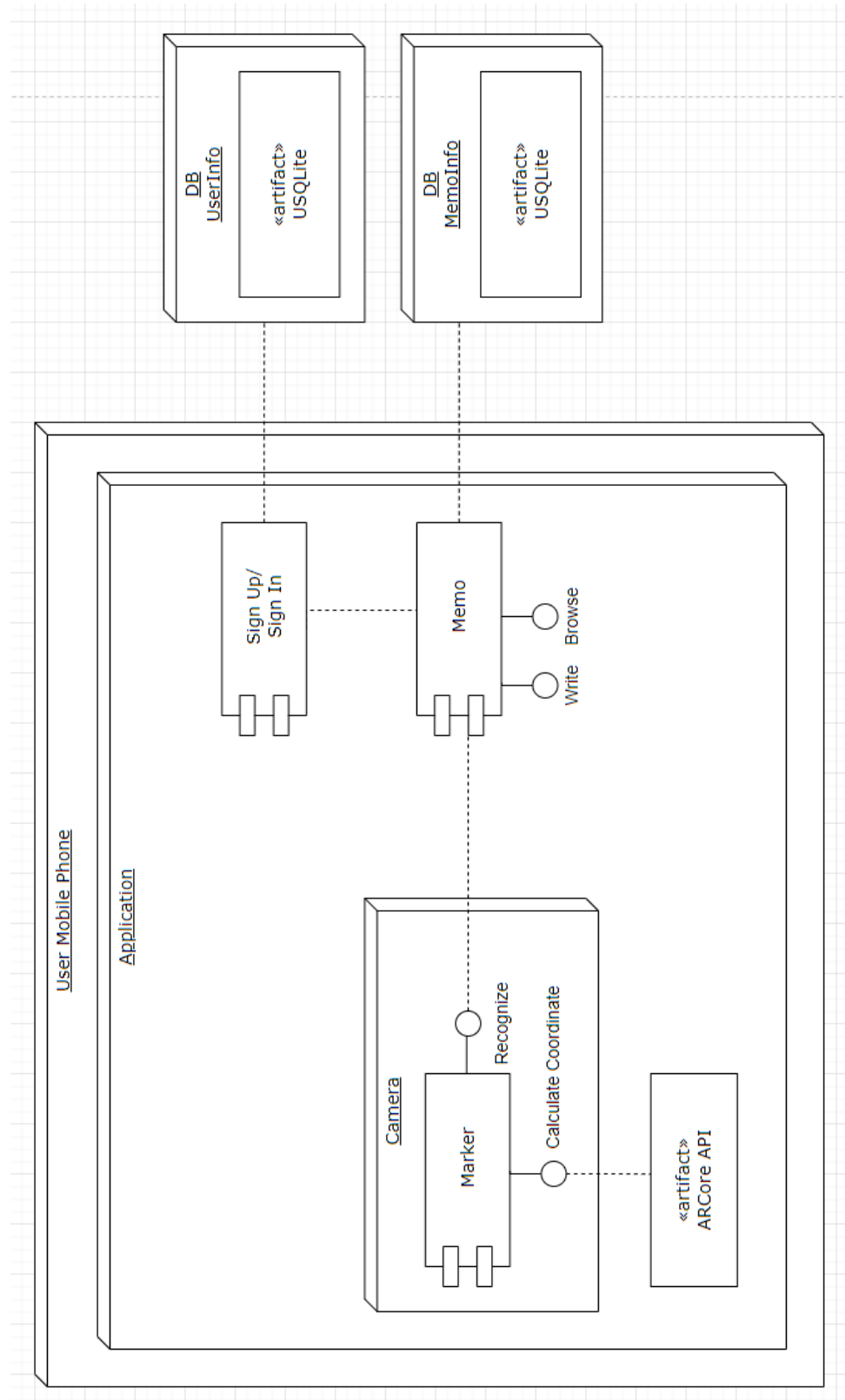
구한 교선의 방정식의 기울기는 x 축의 기울기가 된다. 모바일 폰에 인식된 마커는 원점(0, 0, 0)이 된다. x 축은 기울기가 교선의 방정식이고, 원점인 마커를 지난다. y 축은 x 축의 기울기를 90도 회전한 값이고, y 축도 원점인 마커를 지난다.

위의 방식으로 인식된 마커를 원점으로 x 축과 y 축을 형성하며 좌표계가 만들어진다. 다른 마커가 추가로 인식되면, 인식된 마커는 형성된 좌표계의 좌표를 가진다. 따라서, 원점인 마커와 새로 인식된 마커는 상대적인 좌표를 계산할 수 있게 된다. 마커들 간 상대적인 위치관계를 저장할 수 있게 되면, 애플리케이션을 다시 실행해서 마커들을 재인식했을 때, 저장된 상대적 위치관계를 복구하며 가상메모를 스폰할 수 있다. 상대적인 위치관계를 통해 가상메모를 배치하므로, 가상메모가 범위 밖에서 스폰되는 현상을 개선할 수 있을 것이다.

5. 디자인

6. 절에서 다루는 다이어그램들은 중간 발표 이후에 수정된 부분이 있기 때문에 최종 보고서에서 다루게 되었다.

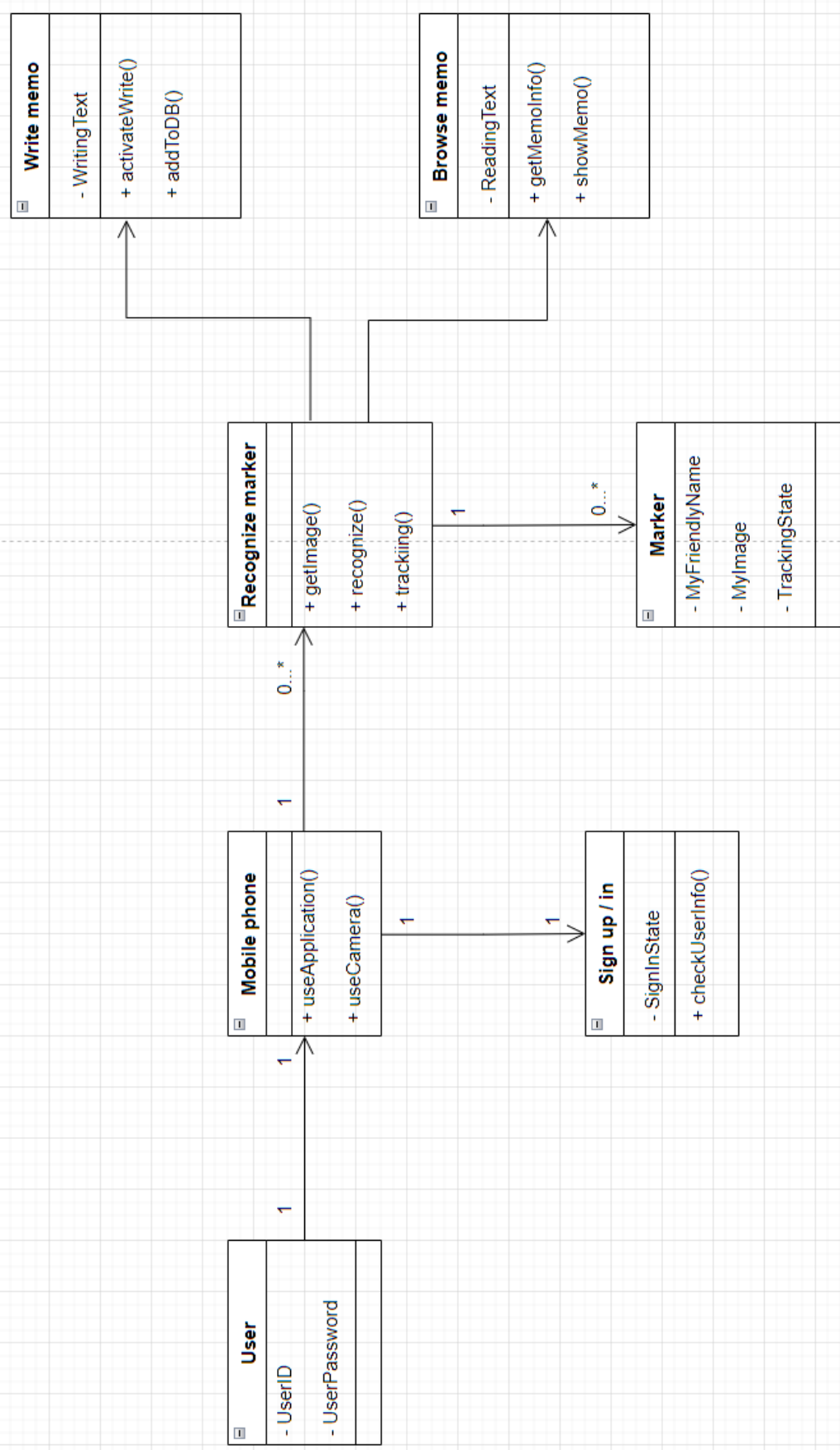
5.1. Deployment Diagram



Deployment Diagram이 중간발표에서와 달라진 부분은 권한에 대한 부분이 삭제되었다는 점과 메모 삭제 기능이 메모 작성 기능에 합쳐졌다는 점이다.

애플리케이션을 실행시킨 사용자가 모바일 폰의 카메라를 이용해 마커 인식을 하면, 마커의 image tracking이 이루어진다. 트래킹 된 마커의 상태에 따라 메모 정보 데이터베이스에 정보를 저장할 수도 있고, 저장된 정보가 제공되기도 한다.

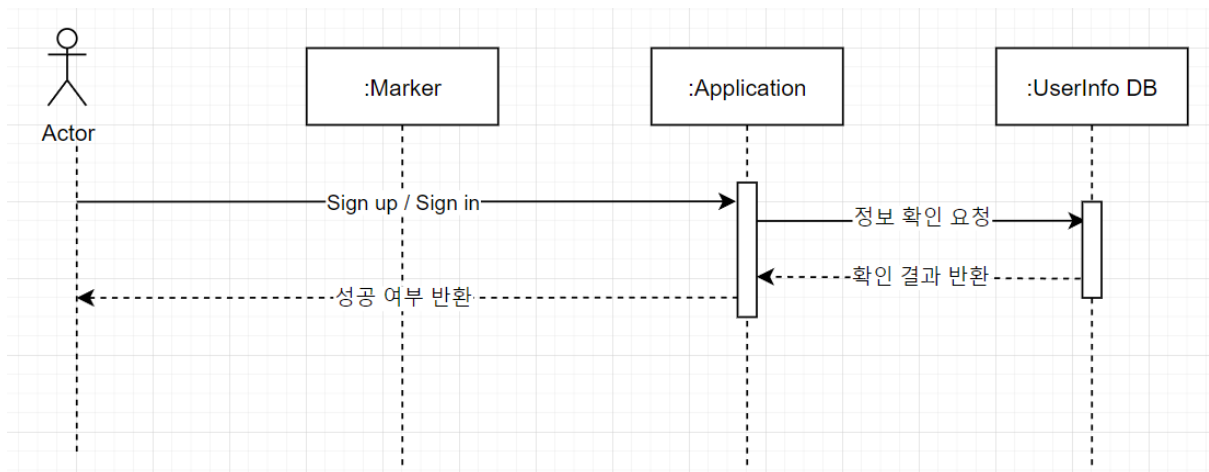
5.2. Class Diagram



클래스 다이어그램에서 중간발표와 비교했을 때 달라진 부분은 권한에 대한 부분이 삭제되었다는 점과 메모 삭제 기능이 메모 추가 기능과 합쳐서 메모 작성 기능으로 바뀌었다는 점이다.

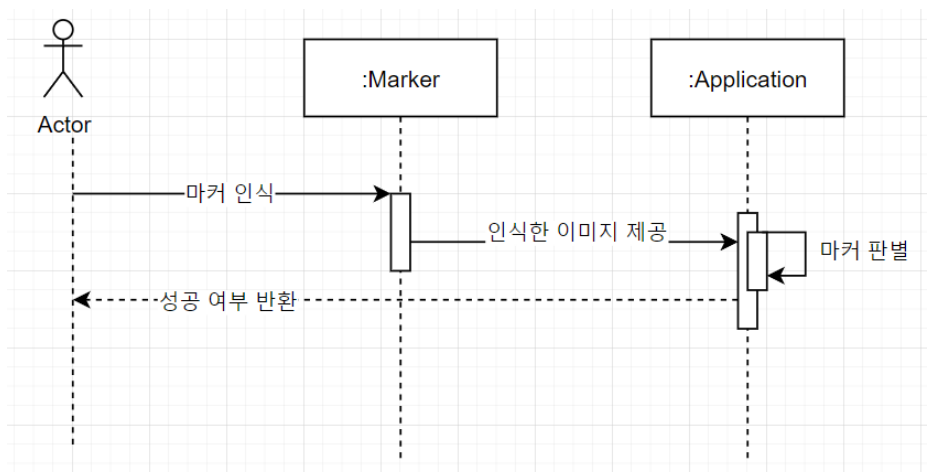
5.3. Sequence Diagram

① Sign up/ in



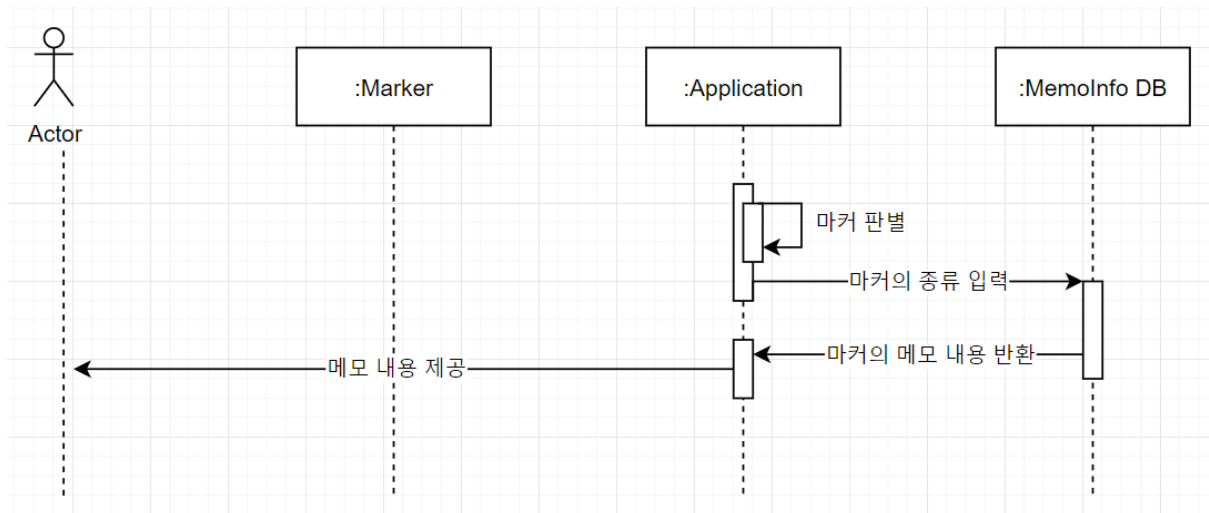
사용자가 로그인을 하면 사용자 정보 DB에서 사용자의 정보를 확인한 후, 로그인 성공 여부를 결정한다.

② Recognize marker



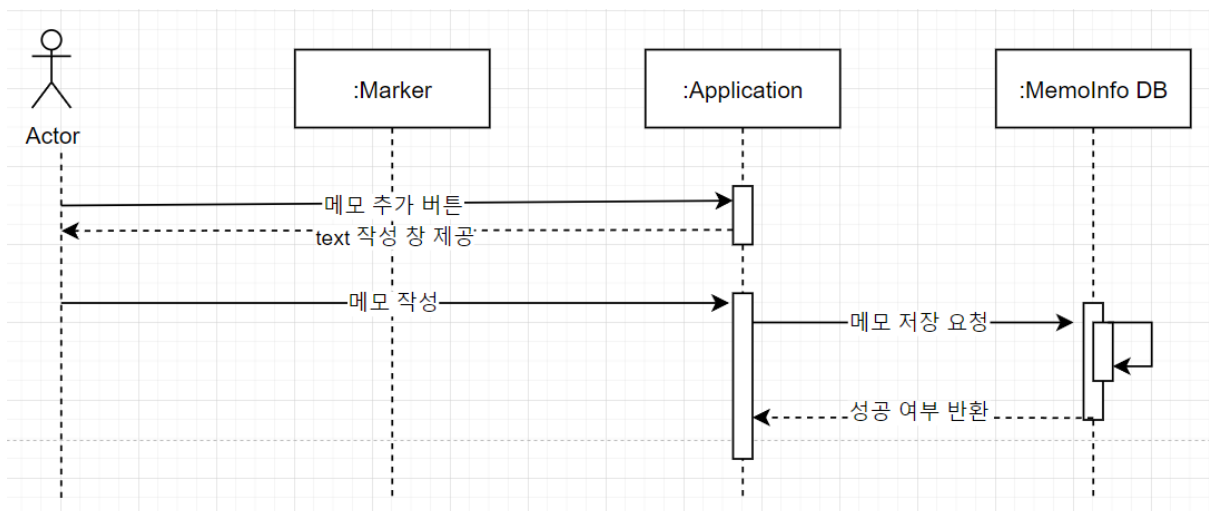
사용자가 이미지를 인식하면, 해당 이미지가 마커일 경우에 어떤 마커인지, 현재 tracking인 중인지 사용자에게 나타낸다.

③ Browse memo



사용자가 마커를 인식했을 때, 저장된 메모가 있다면 메모 정보 데이터베이스에서 정보를 가져와 사용자에게 제공한다.

④ Write memo



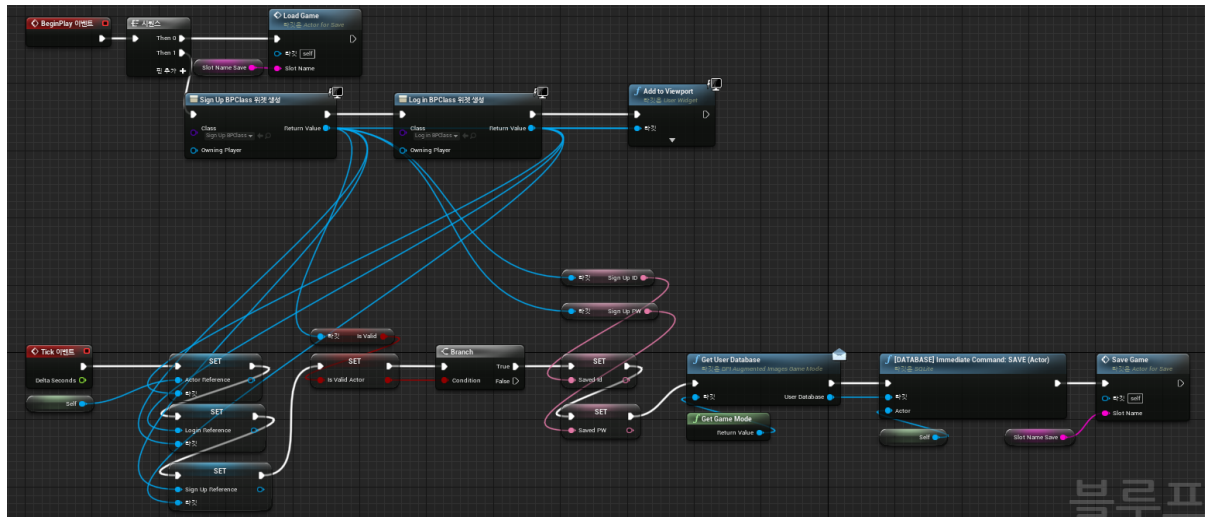
사용자가 Memo 버튼을 누르면, 메모 내용을 작성하는 위젯이 주어진다. 사용자가 작성한 정보는 메모 정보 데이터베이스에 저장된다.

6. 결과물

프로젝트의 노드 흐름을 고려하여 세 파트로 나누었다.

6.1. 회원가입과 로그인

아래 첨부한 사진은 회원가입과 로그인의 흐름을 보여주는 블루프린트 캡처 사진이다. 해당 블루프린트는 프로젝트의 ActorForSave 블루프린트 클래스이다.



애플리케이션이 실행되면 기존에 저장된 데이터를 Load 한다. 회원가입 위젯과 로그인 위젯이 배치된다. 이때, Sign up 버튼을 누르면 로그인 위젯이 hidden 되며 회원가입 위젯으로 이동된다.



사용자가 정보를 작성하고 Confirm 버튼을 누르면 Tick 이벤트가 활성화되며 데이터베이스에 정보가 저장되고, 회원가입 위젯이 hidden되어 로그인 위젯으로 이동한다. 로그인 위젯에서 계정을 입력하면, 회원가입 시 저장한 정보와 같은 지 확인한 후 정보가 일치하면 로그인된다.

① 회원가입

아래의 이미지는 회원가입 위젯이다.



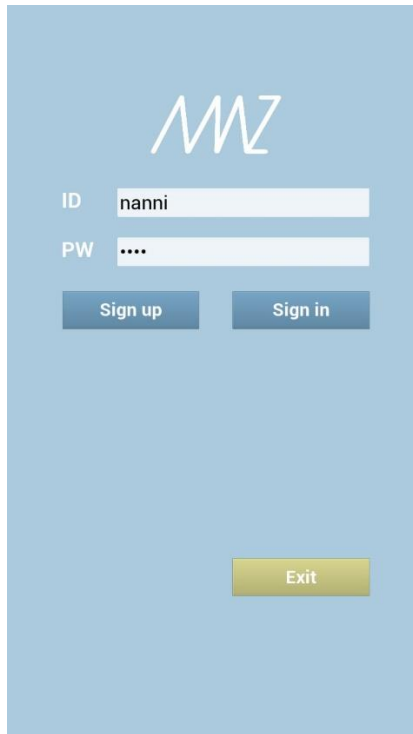
아래는 어플리케이션을 데스크탑에서 시뮬레이션 하였을 때 저장되는 데이터베이스 파일이다.

 Memo_Database	2021-11-23 오후 10:56	Data Base File	12KB
 User_Database	2021-11-25 오후 1:05	Data Base File	12KB

어플리케이션에서도 회원가입에 성공하면, 사진과 같이 데이터베이스 파일이 생성되며 사용자 계정이 저장된다.

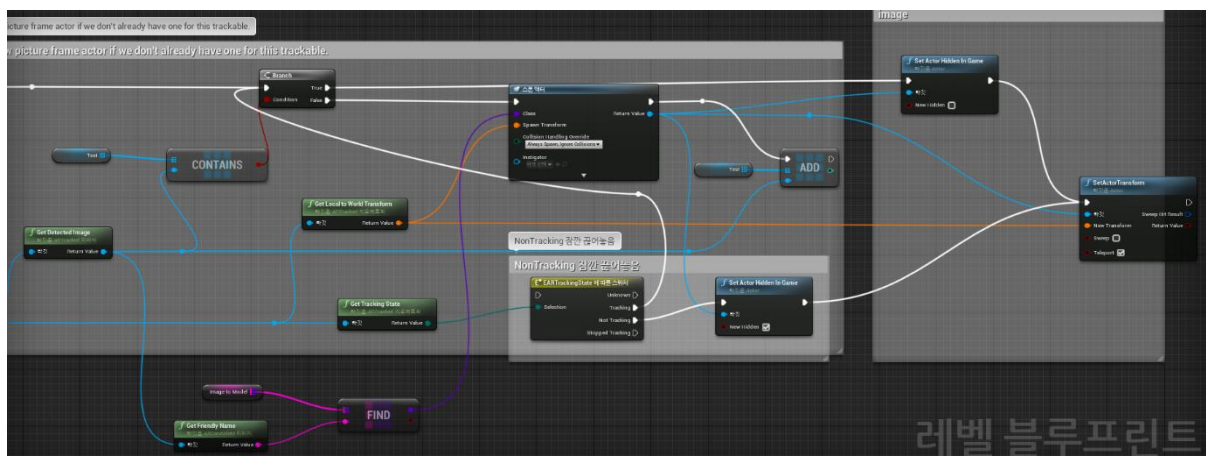
② 로그인

아래의 이미지는 로그인 위젯이다.

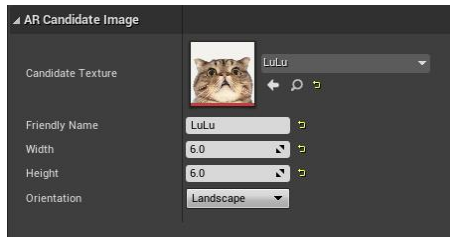


6.2. 마커 인식

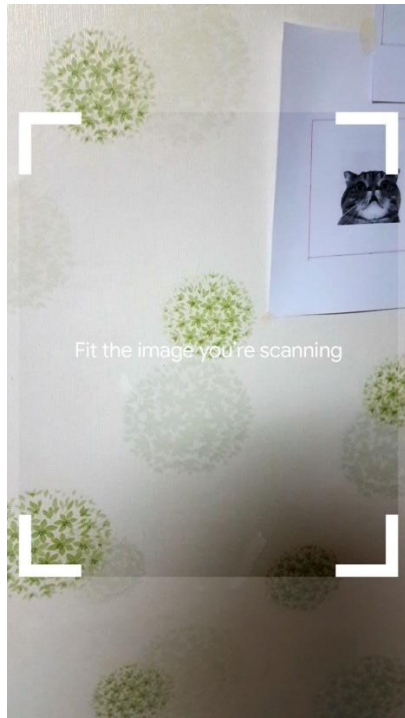
아래 첨부한 사진은 마커 인식에 대한 흐름을 보여주는 블루프린트 캡처 사진이다. 해당 블루프린트는 프로젝트의 레벨 블루프린트이다.



사용자가 모바일 폰의 카메라로 마커를 인식한다. 마커는 아래 첨부된 사진과 같이 AR Candidate Image로 설정하여, 데이터에셋으로 지정하였다. 인식된 마커에 대한 가상메모가 이미 스폰되어 있는지 확인한다. 스폰되어 있을 경우, 새 가상메모를 스폰하지 않고 기존 가상메모의 위치에 대한 업데이트만 이루어진다. 스폰된 가상메모가 없는 경우 마커가 인식된 위치에 새 가상메모를 스폰한다.

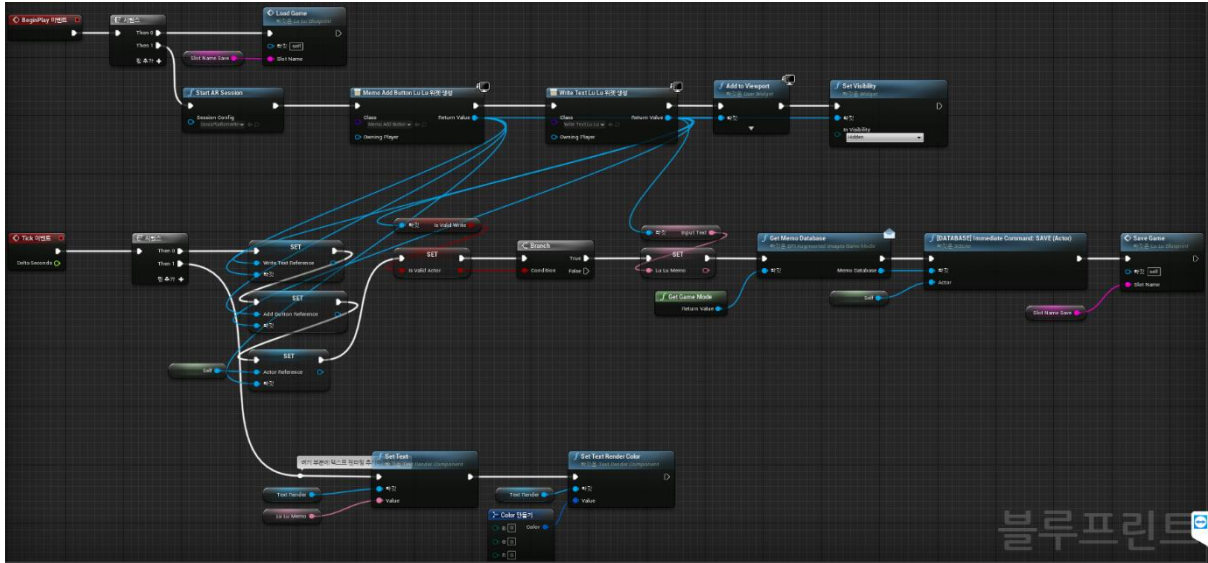


아래의 사용자 인터페이스는 마커 인식을 요구하는 위젯이다.



6.3. 메모 열람과 메모 작성

아래 첨부한 사진은 메모 열람과 메모 작성에 대한 흐름을 보여주는 블루프린트 캡처 사진이다. 해당 블루프린트는 프로젝트에서 각 마커마다 따로 존재하는 블루프린트로, '마커이름_Blueprint'이다.

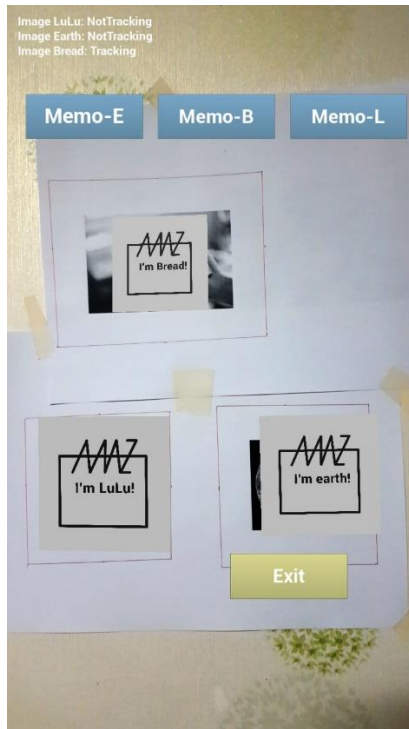


마커를 인식하면, Memo 버튼을 생성하는 위젯이 만들어진다. Memo 버튼을 누르면 메모 작성 위젯으로 이동하며 새 메모를 작성하거나 기존에 작성된 메모를 수정할 수 있다. 메모 작성 위젯에서 메모를 작성한 후 Save 버튼을 누르면 입력한 메모 내용이 메모 정보 데이터베이스에 저장된다.

이후 마커를 인식하는 위젯으로 돌아오며, 가상메모를 통해 저장된 메모 내용을 확인할 수 있다. Tick 이벤트로 메모 정보 데이터베이스에 저장된 메모 내용을 업데이트 하므로, 사용자는 메모 작성 직후 자신이 작성한 메모 내용을 가상메모에서 확인할 수 있다.

① 메모 열람

아래는 가상메모를 통해 메모 내용을 열람할 수 있는 사용자 인터페이스이다.



② 메모 작성

아래는 메모를 작성할 수 있는 위젯이다.



7. Risk Analysis + Risk Reduction Plan

① 극복한 어려움

중간 보고서에서 언급한, 남아있는 어려움에는 데이터베이스의 사용이 있었다. 사용자는 애플리케이션에서 마커를 인식하였을 때, 메모의 내용을 작성/수정할 수 있고, 작성한 메모는 데이터베이스에 저장되어야 한다.

중간보고서 시점에서는 빈 메모를 가진 마커를 인식하였을 때 메모를 작성할 수 있는 기능까지는 구현되었지만, 작성한 메모를 데이터베이스에 저장하는 기능은 구현하지 못했다. 그러나 메모 작성 기능은 프로젝트의 핵심 기능인 만큼, 무조건적으로 구현되어야 하였다.

② 해결 방법

프로젝트를 수행하는 내내 개인적으로 공부를 진행하긴 했지만, 중간 보고서 시점에서 까지 구현에 어려움을 겪었기 때문에 개인적으로 공부하는 것 이상의 수단이 필요하다고 판단하였다. 따라서 Unreal engine의 epic games에서 USQLite Database라는 상용 SW를 구매하여 사용하였다. 해당 SW는 언리얼 엔진 프로젝트에서 사용할 수 있는 플러그인으로, 언리얼 프로젝트 내에서 USQLite가 제공하는 함수를 사용하여 데이터 베이스의 구현을 가능하게 하였다. 해당 SW의 사용으로, 부족했던 배경지식과 시간에도 데이터베이스에 대한 공부와 프로젝트의 구현을 동시에 진행할 수 있었다.

그러나 만들어진 프로젝트를 데스크탑에서 시뮬레이션 하였을 때는 정상적으로 작동하던 데이터베이스가, 모바일 폰에서는 애플리케이션을 종료 후 재접속 하였을 때 저장의 기능을 제대로 수행하지 못하였다. 해당 문제를 극복하기 위해 Save game이라는 언리얼 내 클래스를 사용하였다. Save game을 object로 받는 Save game to slot 함수를 사용하여, 애플리케이션을 설치한 모바일 폰 내부 저장소에 Save game 클래스의 내용을 저장한다. 이 때 저장되는 정보는 사용자 정보와 메모 정보 데이터 베이스인 SQLite data 클래스 파일이다. 이후, Load game을 통해 이전에 저장된 데이터를 읽어왔다.

8. Success Criteria

중간 발표 때 제시한 Primary Use Case에는 마커 인식 기능이 있었다. 중간 발표 때는 다양한 환경에서 여러 마커들이 얼마나 잘 인식되는지 검사하고자 하였다. 그러나 애플리케이션이 실제로 사용되는 환경이 그렇게 다양하지 않다. 또한, 많은 메모를 동시에 제공하는 것이 정보 제공의 측면에서 더욱 효율적이기 때문에 가상메모의 취지에 더 맞다고 생각되어 여러 개의 가상메모를 동시에 스폰하는 것에 대한 실험과 분석을 하는 것으

로 변경하였다.

8.1. 테스트 방법

테스트에서는 카메라로 마커를 인식하기 위한 수단으로 안드로이드 OS를 가진 모바일 폰 Samsung Galaxy S7 edge(SM-G935)를 사용하였다. 테스트를 수행한 장소는 자택에 있는 방이다. 테스트에 사용하는 마커는 세 가지로, 아래와 같다.



테스트는 모바일 폰으로 인쇄한 마커를 인식하며 수행된다. 테스트에서 측정 시 용이하기 위해 모든 마커의 세로 길이는 8센티미터로 동일하게 하여 인쇄한다. 마커를 인식하면 가상메모는 스폰 된다. 가상메모 스폰은 마커의 모서리를 기준으로 가로와 세로 3센티미터까지 오차범위로 허용된다. 오차 범위는 마커를 인쇄한 종이에 붉은 색의 펜으로 표시하여 스폰 성공 여부 판단에 사용한다.

테스트는 세 가지 조건에서 수행된다. 조건 1은 한 가지 마커에 대해 가상메모를 스폰하는 경우, 조건 2는 두 가지 마커에 대해 동시에 가상메모를 스폰하는 경우, 조건 3은 세 가지 마커에 대해 동시에 가상메모를 스폰하는 경우이다.

아래의 표와 같이, 조건 1은 세트 1~3으로, 조건 2는 세트 4~5로, 조건 3은 세트 7로 구성되어 있다. 각각의 세트에 대해서는 100번씩 스폰을 측정한다.

조건	세트	구성 마커	스폰 횟수
조건 1	세트 1	고양이	100
	세트 2	지구	100
	세트 3	빵	100
조건 2	세트 4	고양이, 지구	100
	세트 5	고양이, 빵	100
	세트 6	지구, 빵	100

조건 3	세트 7	고양이, 지구, 빵	100
------	------	------------	-----

모든 조건에서의 테스트 절차는 아래의 표를 따른다.

순서	내용
1	어플리케이션을 실행시켜, 마커 인식 위젯에 도달한다.
2	한 개의 마커 씩 인식한다.
3	조건에서 동시에 스폰하는 가상메모의 개수만큼 순서 2를 반복하여 수행한다. (조건2: 1회 추가 반복/ 조건 3: 2회 추가 반복)
4	모든 가상메모의 스폰이 완료되면, 현재 화면을 캡처한다.
5	가상메모의 스폰 성공 여부를 계산한다.

발생한 가상메모 면적의 70%가 면적 범위 안에 있으면 스폰에 성공한 것으로 간주한다. 단, 조건 2와 조건 3의 경우, 하나의 마커라도 오차 범위를 벗어난다면, 스폰에 실패한 것으로 간주한다.

가상메모가 스폰 되면, 해당 화면을 캡처해서 가상메모 면적의 70%가 오차 범위 안에 존재하는지 확인한다. 마커와 가상메모 모두 사각형이므로, 캡처한 사진에서 범위 밖의 면적을 직사각형으로 나누어서 면적을 계산한다. 전체 가상메모 크기와 비교했을 때 30%가 넘는지 판단한다.

8.2. 테스트 결과

테스트 결과, 세트 1, 세트 2, 세트 3에서의 스폰 성공확률은 각각 100%, 99%, 97%로, 하나의 마커를 인식하였을 경우에는 스폰 성공확률이 높았다. 세트 4, 세트 5, 세트 6에서의 스폰 성공 확률은 각각 72%, 69%, 71%로, 하나의 마커를 인식할 때보다 스폰 성공확률이 감소하였다. 세트 7에서의 스폰 성공확률은 63%이다. 이는 두 개 이하의 마커를 인식할 때보다 스폰 성공확률이 낮으며, 성공확률이 절대적으로 낮다.

즉, 한 번에 스폰하는 마커의 개수가 늘어날수록, 스폰에 실패할 확률이 급격하게 증가한다.

8.3. 테스트 결과 분석

테스트 결과를 통해, 여러 마커에 대해 동시에 가상메모를 스폰하려고 할수록, 스폰 성공확률이 떨어지는 것을 확인할 수 있었다. 해당 문제의 원인은 마커 인식을 통해서만

가상메모를 스폰하였기 때문이라고 분석하였다.

본 프로젝트에서는 마커가 인식되면, 그 위치를 즉석으로 계산해서 가상 메모가 배치된다. 이때, 계산된 가상메모는 로컬 좌표계로 위치가 계산된다. 즉, 인식된 각각의 마커들은 자신의 위치를 로컬 좌표계로 나타낸다. 애플리케이션의 실행에서, 인식된 각 마커들은 가상메모를 배치할 때 독립된 로컬 좌표계를 가진다. 각 마커들은 좌표계를 서로 공유하지 않는다. 즉, 사용자가 하나의 마커를 인식해서 그 로컬 좌표계에 가상메모를 배치하고, 다른 마커를 새로 인식하면 새 로컬 좌표계를 생성해서 가상메모를 배치한다. 이때, 사용자가 새 마커 인식을 위해 모바일 폰을 빠르게 움직이면, 현재 인식하려는 위치가 이전 마커 인식을 통해 인식된 위치에서 크게 달라지지 않았다고 인식하여, 이전 마커의 좌표에 가상메모를 배치하는 문제가 발생한다고 분석하였다.

다른 마커의 좌표계가 간섭한 것을 문제 원인으로 생각한 이유는, 하나의 마커에 대해서만 가상메모를 배치할 때는 가상메모가 의도하지 않은 위치에 발생하는 경우가 거의 없었기 때문이다. 즉, 인식된 마커들이 공유하는 하나의 좌표계가 만들어진다면, 각 마커의 로컬 좌표계 간섭으로 인한 문제가 개선될 수 있을 것이다.

9. 개선사항

프로젝트의 개선을 위해, 구현하지 못한 기능을 여건이 된다면 추후에 추가하고자 한다. 추가하고자 하는 기능은 아래와 같다.

① 메모 열람에 대한 권한

메모 열람 권한에 대한 기능은 데이터베이스 사용 미숙으로 시간관계상 구현하지 못하였다. 해당 기능은 프로젝트에서 기술적으로 핵심이 되는 기능은 아니지만, 프로그램의 정체성에 추가 되는 기능이라고 생각하여 추후에 구현하고자 한다.

② 3차원 위치 좌표계 형성과 복구

3차원 위치 좌표계 형성과 복구 방법에 대한 고안은 nnnn절에 제안되어 있다. 현 프로젝트에서 사용한 Unreal Engine 4.26.2로 제안한 방법을 구현하고자 한다.

10. 참고자료

- ① Park, J. H., 「Exploring Augmented Reality applications for Geography Learning:

Focused on Marker Based Methods」, 『대한지리학회지』, 48(6): 994~1008, 2013.

- ② Jung, J. J., 「A Study on Stable Service of Marker based Augmented Reality Using 3D Location Measurement of Beacons」, 『Journal of the KIECS, 12(5): 883~890, 2017