

1. CafeIN 앱 개발 설계

개발내용	개발방법	구성 환경
카페 정보 제공 서비스	<p><유저(User) 기능></p> <ul style="list-style-type: none"> - 카페의 주소, 영업시간, 별점 -> 목차를 통한 카페 정렬 - 카페 및 메뉴 검색 기능 - 카페 홍보 배너를 통한 광고 서비스 	Unity를 통한 개발, 데이터 베이스 구축
카페 위치 제공 서비스	<p><유저(User) 기능></p> <ul style="list-style-type: none"> - 개발에 사용 가능한 MapBox를 통해 카페 위치 제공 - 카페 위치 정보에 대한 마커 제공 	Unity, MapBox
카페 잔여석 제공 서비스	<p>방법 1) 직접 좌석 업데이트</p> <ul style="list-style-type: none"> - 오너(Owner)앱을 통해 오너가 직접 실시간 잔여석 업데이트 - 실제 카페의 좌석 위치 맵 제공 -> 오너 전용 어플을 통해 좌석 배치 시스템 개발 	Unity
	<p>방법 2) IoT 좌석벨 개발</p> <ul style="list-style-type: none"> - 방법 2에 대한 개발 어려움이 있을 경우 방법 3으로 시행 - 사물인터넷을 이용한 좌석벨 개발을 통해 카페 좌석 위치에 대한 실시간 잔여석 업데이트 	아두이노 버튼 모듈, IoT 시스템 개발
카페 좌석 예약 서비스	<p><유저(User) 기능></p> <ul style="list-style-type: none"> - 카페 좌석 위치 맵에서 전화를 통해 사용자 좌석 예약 가능 	Unity

2. 설계한 내용의 구현 방법 및 과정

1) 아두이노 센서를 활용한 IoT 좌석벨 제작

[아두이노 개발 과정]

1. 아두이노 우노를 PC에 연결

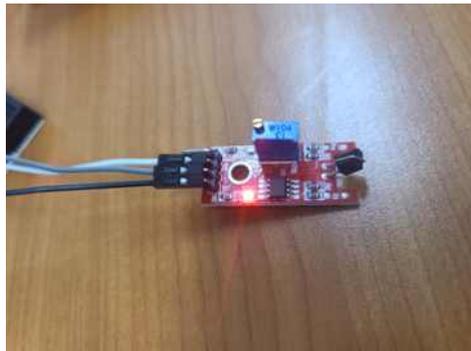


2. 원활한 코딩을 위해 아두이노 IDE를 설치 후 실행

3. 터치 센서 준비



TTP223B 터치센서 모듈을 사용하려 했으나 아직 구하지 못함
대체품으로 KY-036를 사용함



(전원이 연결되어

LED가 점등된 것이 보인다)

뒷부분 검은색 센서 부분을 터치하면 시리얼 모니터에 인식이 된다.

4. 센서를 터치할 시 LED에 점등이 되는 코드를 입력한다.

```

Blink | 아두이노 1.8.13
파일 편집 스케치 볼 도움말
Blink5
int touchSensor = 8; // 터치센서 핀 설정
int ledPin = 13;     // LED 핀 설정

int state = LOW;    // LED 상태
int reading;       // 터치센서 상태
int previous = LOW; // 터치센서 이전 상태

void setup()
{
  pinMode(touchSensor, INPUT); // 터치센서 볼 설정
  pinMode(ledPin, OUTPUT);     // ledPin 설정
}

void loop()
{
  reading = digitalRead(touchSensor); // touchSensor 상태 읽음

  if (reading == HIGH && previous == LOW) {
    if (state == HIGH) // ledPin 가 HIGH 면 LOW 로 바꿔준다.
      state = LOW;
    else // ledPin 가 LOW 면 HIGH 로 바꿔준다.
      state = HIGH;
  }

  digitalWrite(ledPin, state);

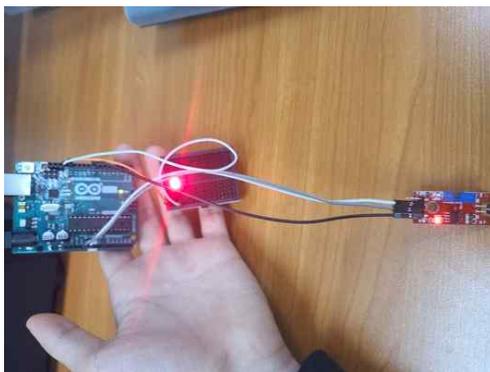
  previous = reading;
}

```

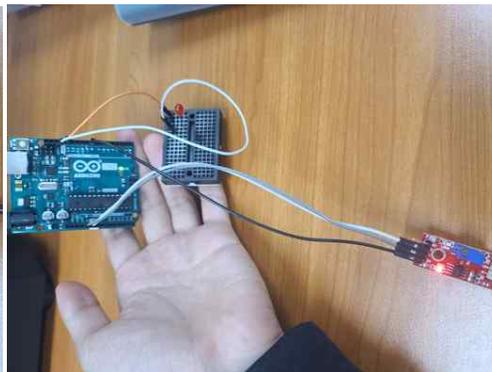
스케치는 프로그램 저장 공간 994 바이트 (3%)를 사용. 최대 32256 바이트.
 지역 변수는 동적 메모리 15바이트 (0%)를 사용, 2033바이트의 지역변수가 남음. 최대는 2048

30 Arduino Uno en COM3

5. 잘 작동이 된다.



(터치 시)



(한번 더 터치 시)

▶ 참조 사이트: <https://deneb21.tistory.com/384>

터치 센서가 쉽게 구현되어 버튼 스위치 모듈도 한번 도전하였다.

1. 스위치 센서



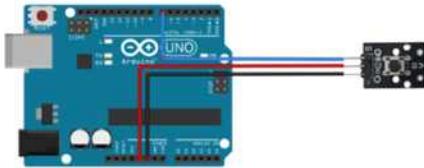
PIN	이름	기능
1	-	GND
2		5V
3	S	Signal (디지털 출력)

버튼스위치 모듈은 일반 탭트 스위치로

구성되어있는 스위치 모듈이다.

버튼을 누르면 스위치가 닫히게 되고 손을 떼면 스위치가 열리게 되어있다.

2. 연결도



3.소스코드

```

//*****
//*****
// Project : 22_BUTTON in Sensor Kit
// Board : Arduino Uno
// By : Kit Plus
//*****
//*****
//int AnalogPin = A0: // Analog input
int DigitalPin = 7; // Digital input
int LedPin = 13; // select the pin for the LED
int gValue = 0; // variable for Sensor Data
void setup ()
{
  pinMode (LedPin, OUTPUT);
  pinMode (DigitalPin, INPUT);
  Serial.begin (9600);
}
void loop ()
{
  gValue = digitalRead(DigitalPin);

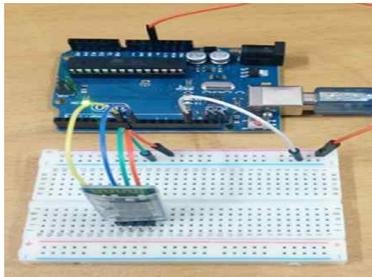
  if (gValue){
    digitalWrite (LedPin, HIGH);
  }else {
    digitalWrite (LedPin, LOW);
  }
  Serial.print ("Sensor Value : ");
  Serial.println (gValue, DEC);
  delay(500);
}

```

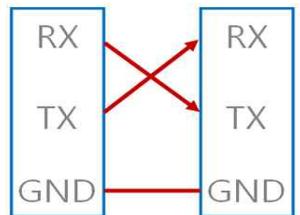
4. 결과 창



여기서 주의할 점은 TXD, RXD 연결시 아두이노와 블루투스 모듈을 반대로 연결합니다.
 즉, 코드에서 입력한 RX, TX 핀번호는 아두이노의 핀번호이며, 아두이노의 RX 핀은 블루투스 모듈의 TX에 연결하고, 아두이노의 TX 핀은 블루투스 모듈의 RX에 연결합니다.



[참고사항] 블루투스 모듈 연결하기



```

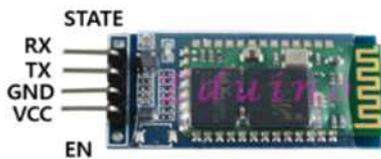
1 #include <SoftwareSerial.h>
2
3 #define BT_RXD 8
4 #define BT_TXD 7
5 SoftwareSerial bluetooth(BT_RXD, BT_TXD);
  
```

[아두이노 블루투스 개발]

1. 블루투스 hc-06이란?

▶ 블루투스는 1994년에 최초로 개발된 근거리 무선 통신을 위한 산업 표준입니다. HC-06은 아두이노에서 시리얼통신을 이용하여 데이터 값을 주고 받을 수 있는 모듈이며, 10미터 정도에서 무선으로 데이터를 보내거나 받을 수 있습니다.

※HC-06모듈의 경우 아이폰(iOS)에서 사용이 불가능합니다.



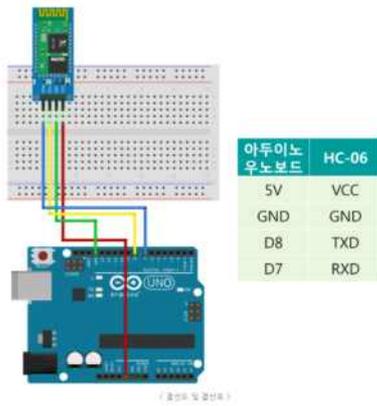
<HC-06 핀 구성>

1. HC-06을 이용한 시리얼 모니터로 AT명령어 확인하기

▶ HC-06를 연결 후 시리얼모니터로 간단한 AT 명령어를 확인합니다.

2. 아두이노와 연결하기

아래 결선도와 같이 아두이노와 HC-06를 연결합니다.



```
#include <SoftwareSerial.h>

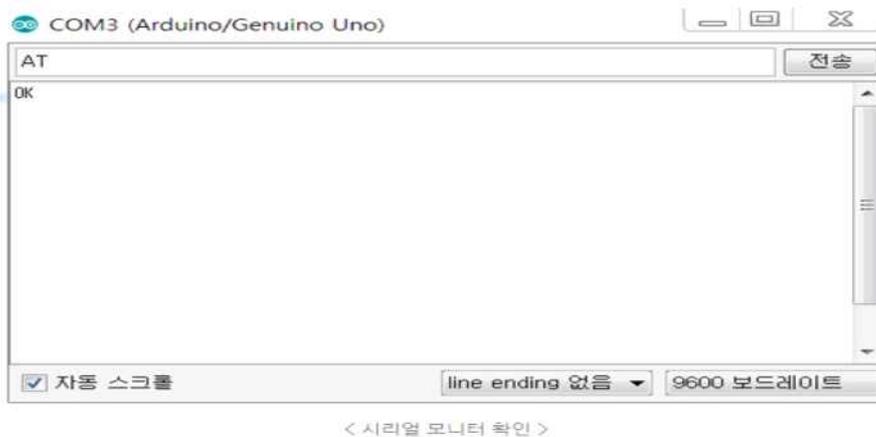
#define BT_RXD 8
#define BT_TXD 7
SoftwareSerial bluetooth(BT_RXD, BT_TXD);

void setup(){
  Serial.begin(9600);
  bluetooth.begin(9600);
}

void loop(){
  if (bluetooth.available()) {
    Serial.write(bluetooth.read());
  }
  if (Serial.available()) {
    bluetooth.write(Serial.read());
  }
}
```

3. 코딩

4. 센서 테스트



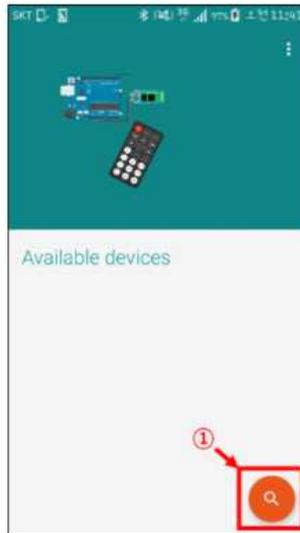
▶ 시리얼 모니터에서 'AT' 입력 시 'OK' 라고 올라오는 것을 확인할 수 있습니다.

5. 스마트폰 어플과 블루투스 통신하기

- ▶ 블루투스 어플을 이용해 스마트폰과 HC-06의 통신을 해봅니다.
- ▶ HC-06 회로구성 및 소스코드 업로드까지 마친 후 스마트폰의 어플스토어에서 'Arduino Bluetooth Controller' 이름을 가진 어플리케이션을 다운로드 및 설치해줍니다.



< Arduino Bluetooth Controller 앱 설치 >

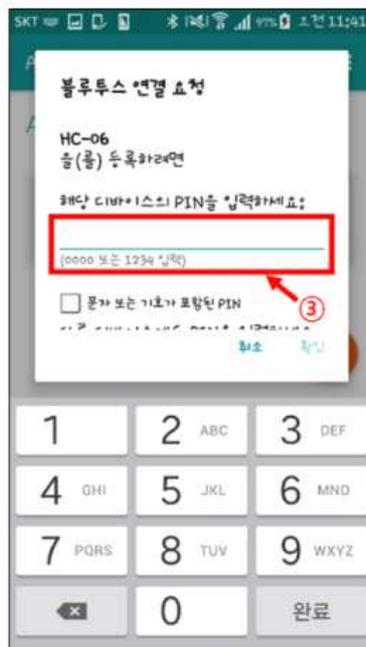


< PIN번호 입력 >

▶ 위 그림과 같이 먼저 스마트폰에 설치해둔 'Arduino Bluetooth Controller' 어플을 실행합니다. 어플 실행 후 그림의 ①번을 터치하면 스마트폰 주변의 블루투스 장치 검색을 시작합니다.

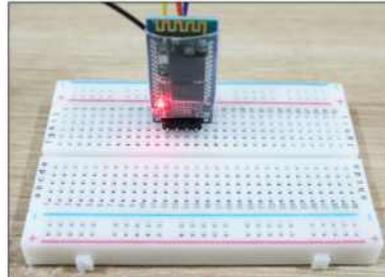


< 주변 블루투스 장치 검색결과 >



< PIN번호 입력 >

▶ 장치 검색시 약 3~5 초 정도가 지나면 위 그림과 같이 HC-06 이 검색됩니다. (주변 에 기타 다른 블루투스 장치가 있으면 같이 검색될 수 있습니다.) 검색된 HC-06 을 터치하면 오른쪽 그림과 같이 '해당 디바이스의 PIN 을 입력하세요:'라는 문구와 함께 창이 뜹니다. 초기 PIN 번호는 **1234** 를 입력해줍니다. PIN 번호를 입력하면 '확인' 버튼이 활성화되며 이를 터치해줍니다.



< HC-06과 어플 연결 확인 > < HC-06의 LED가 켜진상태로 유지 >

▶ PIN 번호 입력까지 정상적으로 마치면 그림과 같이 연결 모드를 선택하는 화면이 나옵니다. 모드는 총 4 가지 선택이 가능하며, 여기서는 ' Terminal mode '를 선택하여 스마트폰과 블루투스 모듈 간의 문자 송수신을 해보겠습니다.
(Terminalmode 를 터치하면 HC-06 과 스마트폰이 연결되며, HC-06 의 LED 가 켜진상태로 유지됩니다.)

▶ 어플에서 'Hello, edduino! - Nice to meet you' 순서로 입력후 전송을 해보면 아두이노 시리얼 모니터에서 문자 그대로 전송됨을 확인할 수 있습니다.



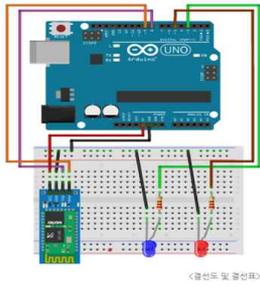
< 시리얼 모니터 확인 >

6. 블루투스를 이용하여 LED 제어하기

▶ 블루투스 어플을 이용해 스마트폰으로 LED 제어를 해보았습니다.

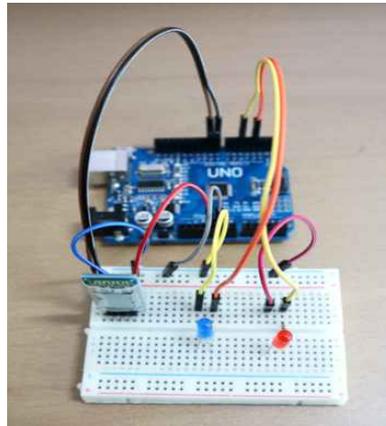
6.1 아두이노와 연결하기

▶ 아래 결선도와 같이 HC-06, LED, 저항을 연결합니다.



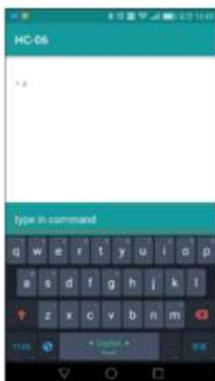
우노 보드	LED
D3	파랑 +
D5	빨강 +
GND	GND

우노 보드	HC-06
VCC	VCC
GND	GND
D8	TX
D9	RX

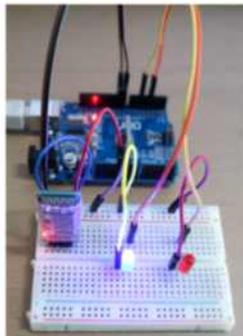


6.2 코딩 예제

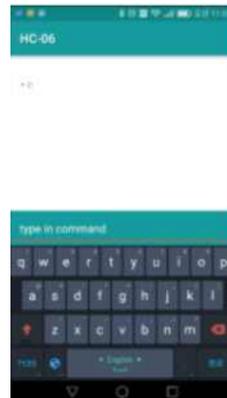
▶ 센서와의 연결을 마친 후 아두이노와 PC를 연결하여 아래 코드를 아두이노 스케치에 입력 후 업로드 해줍니다.



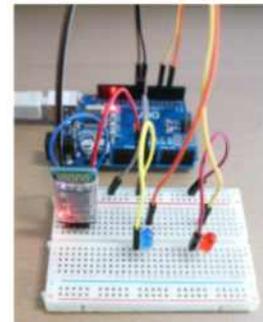
< 어플에서 'a' 입력 >



< 입력 후 파란색 LED 점등 >



< 어플에서 'c' 입력 >



< 'c' 입력 후 LED 소등 >

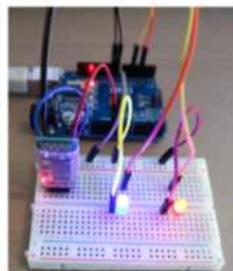
▶ 'Terminal mode' 에서 'a'를 입력하면 위 사진처럼 파란색 LED 점등 되는 것을 확인할 수 있습니다.

▶ 'Terminal mode' 에서 'b'를 입력하면 위 사진처럼 붉은색 LED 점등 되는 것을 확인할 수 있습니다.

▶ 'Terminal mode' 에서 'c'를 입력하면 위 사진처럼 모든 LED 를 소등합니다.



< 어플에서 'b' 입력 >



< 입력 후 붉은색 LED 점등 >

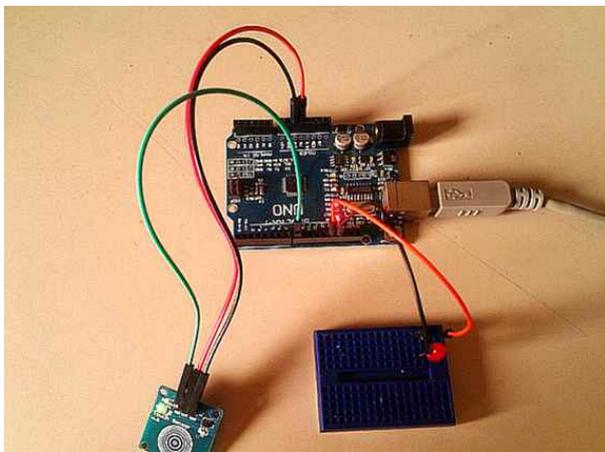
★제일 먼저 터치 센서를 입력 시 LED가 켜지고 꺼질 수 있게 구현

터치의 감지는 예전의 PDA나 네비게이션 같은데 많이 쓰이던 감압식과 요즘 스마트폰에서 쓰이는 정전식 방식이 있음. 감압식은 손가락이나 터치펜 등으로 누르는 압력을 감지하여 작동하는 방식.

정전식은 손의 전류를 감지해서 작동하는 방식. 요즘은 거의 정전식이 많이 쓰임.
현재 사용할 센서도 몸에 미세하게 흐르는 전류를 감지해서 작동하는 터치센서.



터치감지부에서 감지하면 Signal 단자로 HIGH 신호를 보내도록 되어 있음.
전원은 5V를 사용. 하지만 테스트 결과 3.3V 에서도 무난히 작동.



Touch Sensor	Arduino Uno
Signal	D8
VCC	5V or 3.3V
GND	GND
LED	Arduino Uno
Anode, 다리 긴 쪽	D13
Cathode, 다리 짧은 쪽	GND

```

1 int touchSensor = 8; // 터치센서 핀 설정
2 int ledPin = 13; // LED 핀 설정
3
4 int state = LOW; // LED 상태
5 int reading; // 터치센서 상태
6 int previous = LOW; // 터치센서 이전 상태
7
8 void setup()
9 {
10 pinMode(touchSensor, INPUT); // 터치센서 핀 설정
11 pinMode(ledPin, OUTPUT); // ledPin 설정
12 }
13
14 void loop()
15 {
16 reading = digitalRead(touchSensor); // touchSensor 상태 읽음
17
18 if (reading == HIGH && previous == LOW) {
19 if (state == HIGH) // ledPin 가 HIGH 면 LOW 로 바꿔준다.
20 state = LOW;
21 else // ledPin 가 LOW 면 HIGH 로 바꿔준다.
22 state = HIGH;
23 }
24
25 digitalWrite(ledPin, state);
26
27 previous = reading;
28 }

```

손가락이 터치되면 LED가 켜지고 다시 한 번 터치하면 꺼지는 소스

[안드로이드와 아두이노 블루투스 연결]

● 진행과정

1. 아두이노에 HC-06블루투스 모듈 설치
2. 아두이노 세팅 및 코딩
3. 유니티프로젝트 생성 및 세팅
4. 안드로이드 빌드 및 블루투스 연결

1. 아두이노에 HC-06모듈 및 마그네틱 센서 설치

HC-06 : 디지털인풋 2/3번/GND/5V 에 연결.

터치 센서 : 디지털인풋 8번/GND에 연결.

블루투스 모듈을 슬레이브로 변경하기 위해

시리얼 모니터 창을 열어 명령어를 입력.

AT+ROLE=S (M입력시 마스터 로 변경됨)

시리얼 모니터창에서 "AT"라고입력하여 OK가 반환되면,블루투스 모듈 연결이 잘 된 것.

AT+NAME이름을 입력하면 블루투스 모듈 이름을 설정가능.

모두 설정되었다면 아두이노에 코딩작업 시작

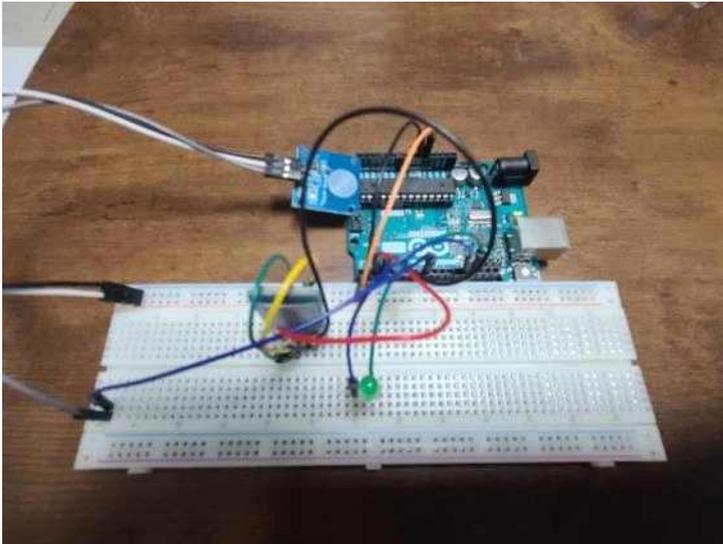
위에서 만들었던 LED코드에 터치센서 코드를 결합

=====

코드를 대략 요약하자면 터치 센서가 클릭되면 "on", 떨어지면 "off"라는 스티링을 블루투스 시리얼로 보내는 동작

[제품의 최소화와 카페 모델 제작]

앞에 만들었던 카페 좌석 알림 벨 ver1의 모습이다.

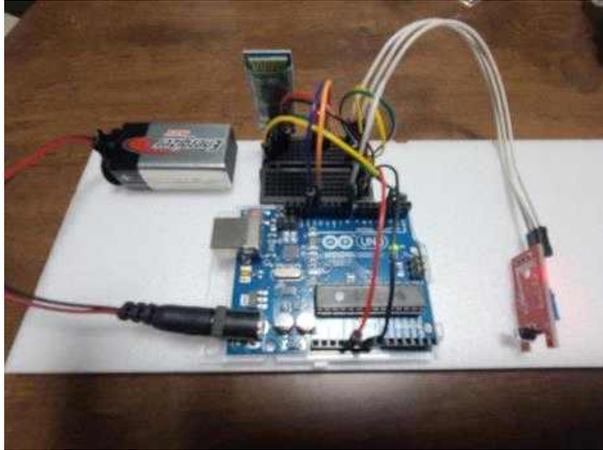


브레드보드가 지나치게 크고 선 정리가 안 되어 있어

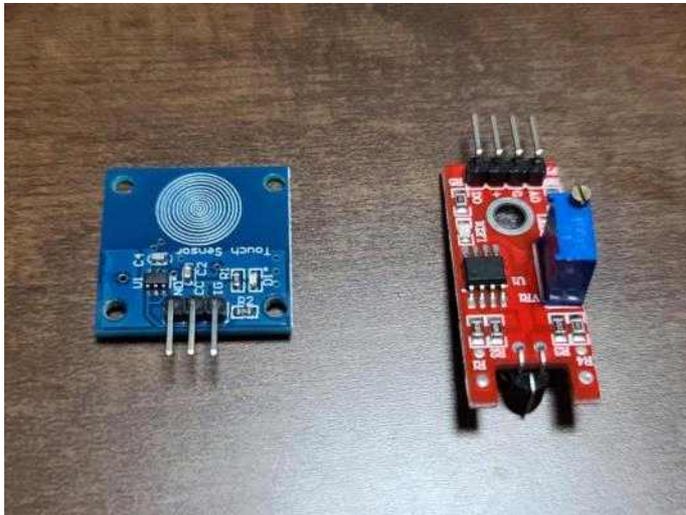
1. 브레드포트 최소화
2. 선 정리 후 포트 색 맞추기.
3. 건전지와 연결하기

위 세가지를 진행하였다.

아래 ver2의 모습이 나왔다.

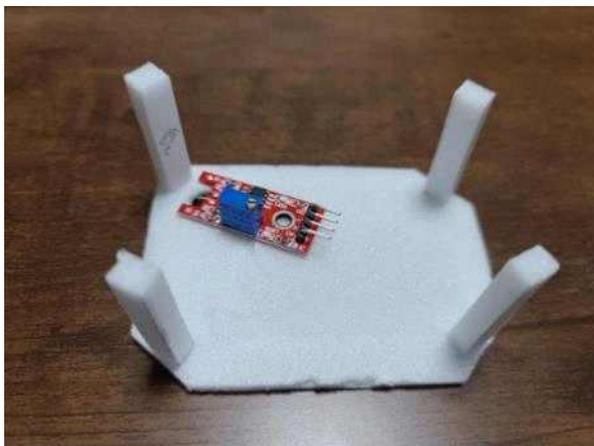


이제 우드락으로 카페 모형을 만들어 벨의 동작을 시연할 것이다.
벨은 가진 터치 센서를 모두 사용하여 두 가지 타입을 만들 것이다.

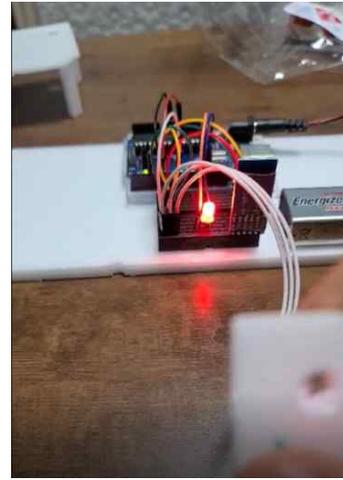


위에서 소개했던 서로 다른 작동 방식의 두 가지 터치 센서이다.

1. 먼저 금속 터치 센서 모듈인 KY-036을 사용하여 테이블을 만들어 보았다.

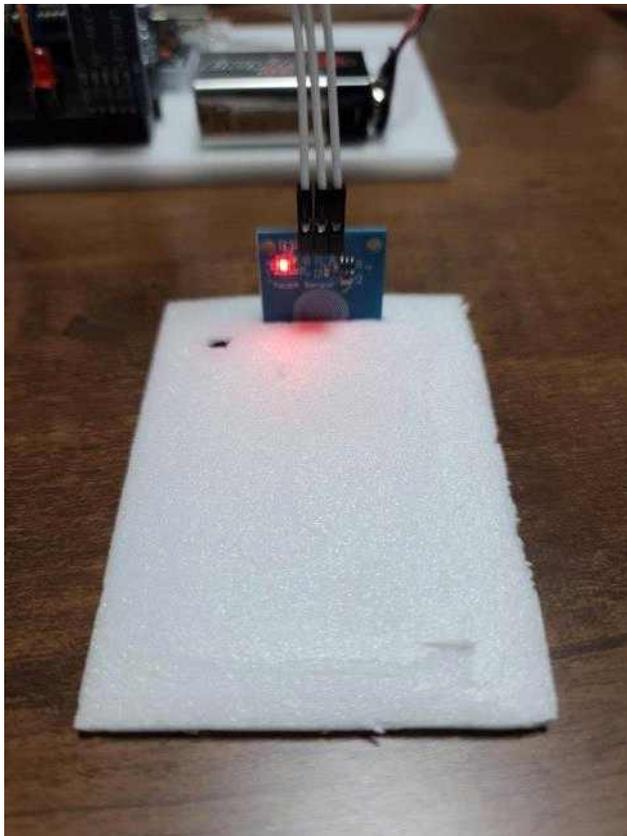


위에서 본 모습이다. 제법 벨 같은 모습이 나온다.



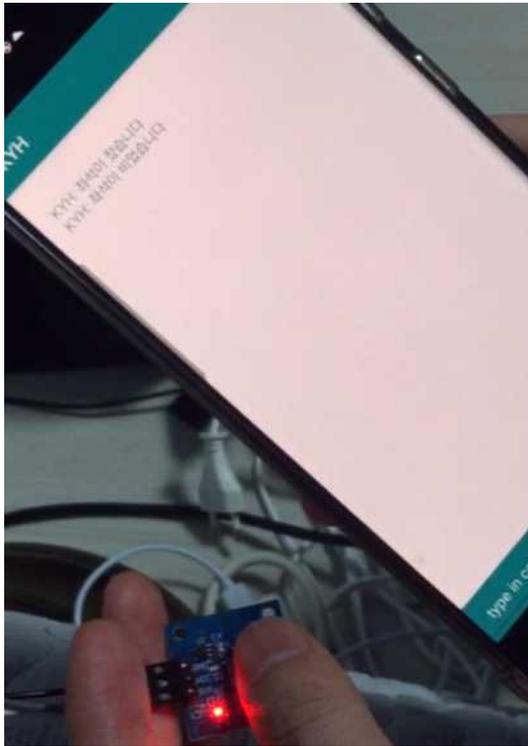
이제 튀어나온 금속을 누르게 되면 영상과 같은 모습이 나온다.

2. 정전식 터치센서 TTP223B를 사용한 테이블을 만들어 보았다.



테이블이 될 우드락 및 부분에 정직하게 끼운 모습이다.

테이블 윗부분 튀어나온 것에 스치기만 하여도 인식이 된다.



KYH: 좌석이 있습니다
KYH: 좌석이 비었습니다
KYH: 좌석이 있습니다
KYH: 좌석이 비었습니다
KYH: 좌석이 있습니다
KYH: 좌석이 비었습니다

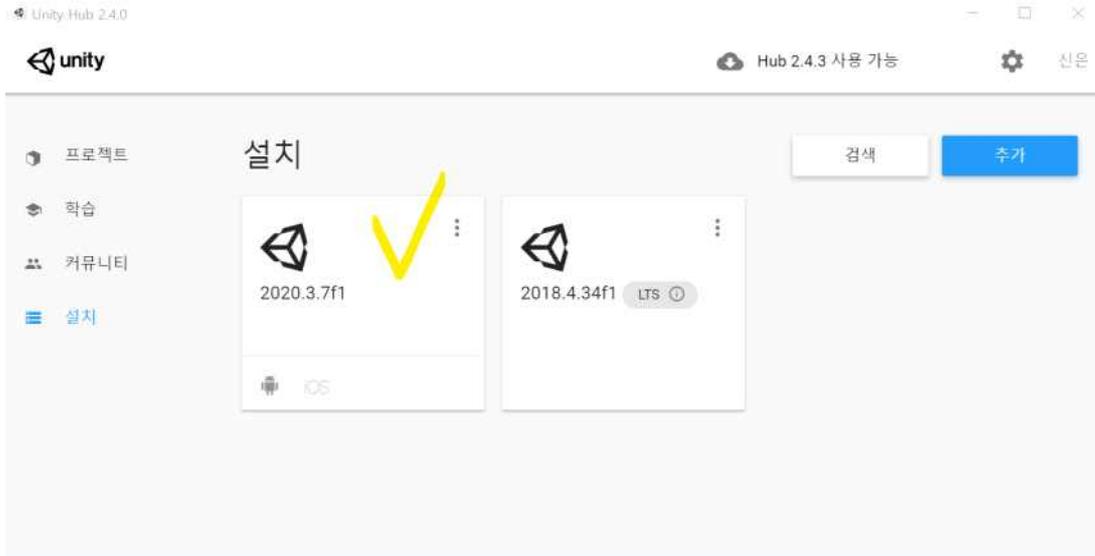
(터치시 인식이 되는 모습)
안드로이드와 블루투스로 잘 신호를 받는다.



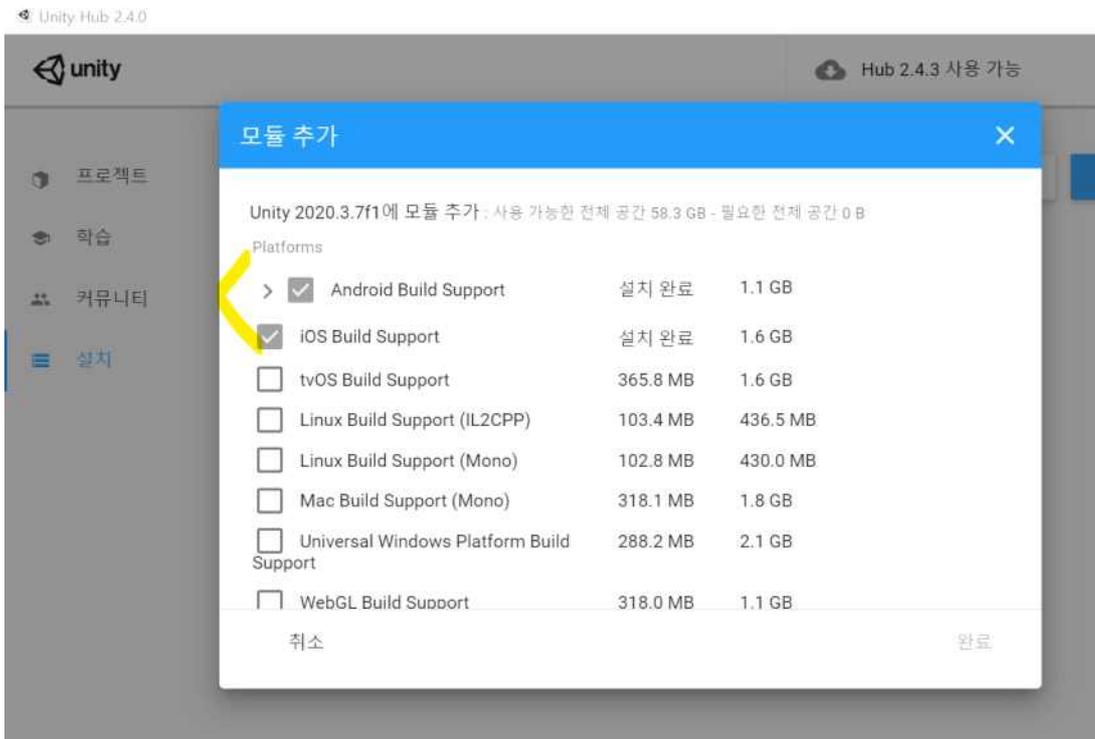
최종 카페 시뮬레이션 모델링 완성

2) 카페 좌석 앱(CafeIN) 개발

<Unity 개발 환경 설치>



- Unity Hub에서 설치>2020 버전을 설치함 -> 오른쪽 위에 추가버튼으로 설치함

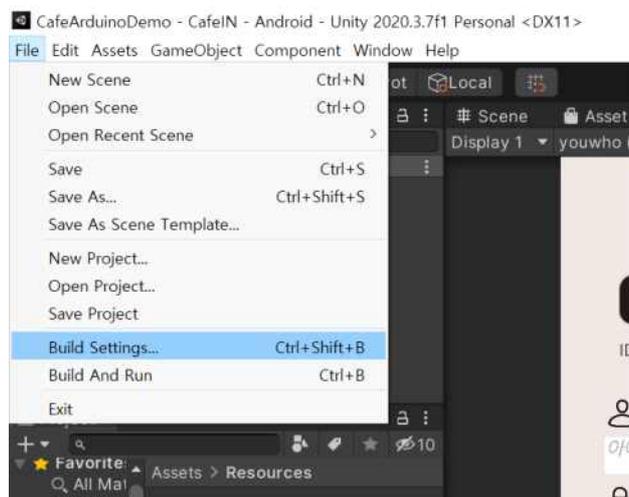


- 설치버튼을 누르면 Android 모듈을 체크해 추가함

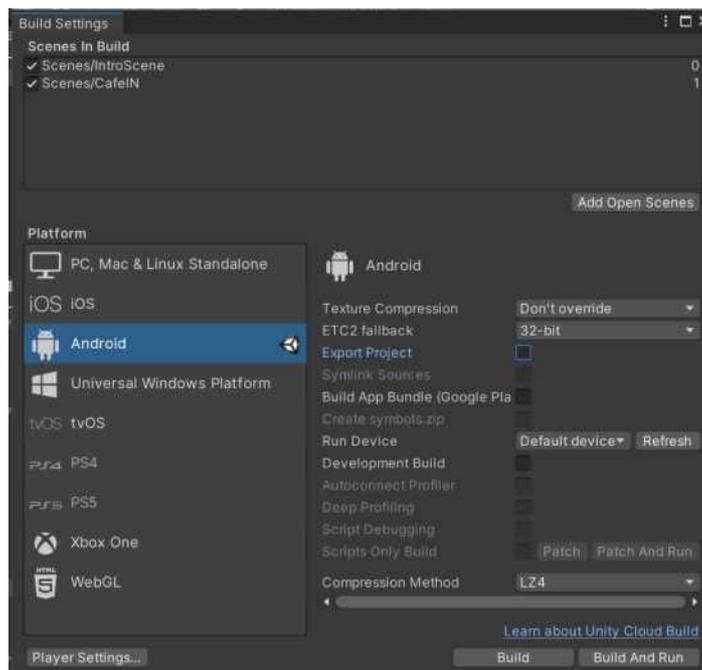


- 설치가 끝나면 추가버튼을 통해 프로젝트 파일을 불러오고 버전을 맞춰줌

<Unity에서 안드로이드 설정>

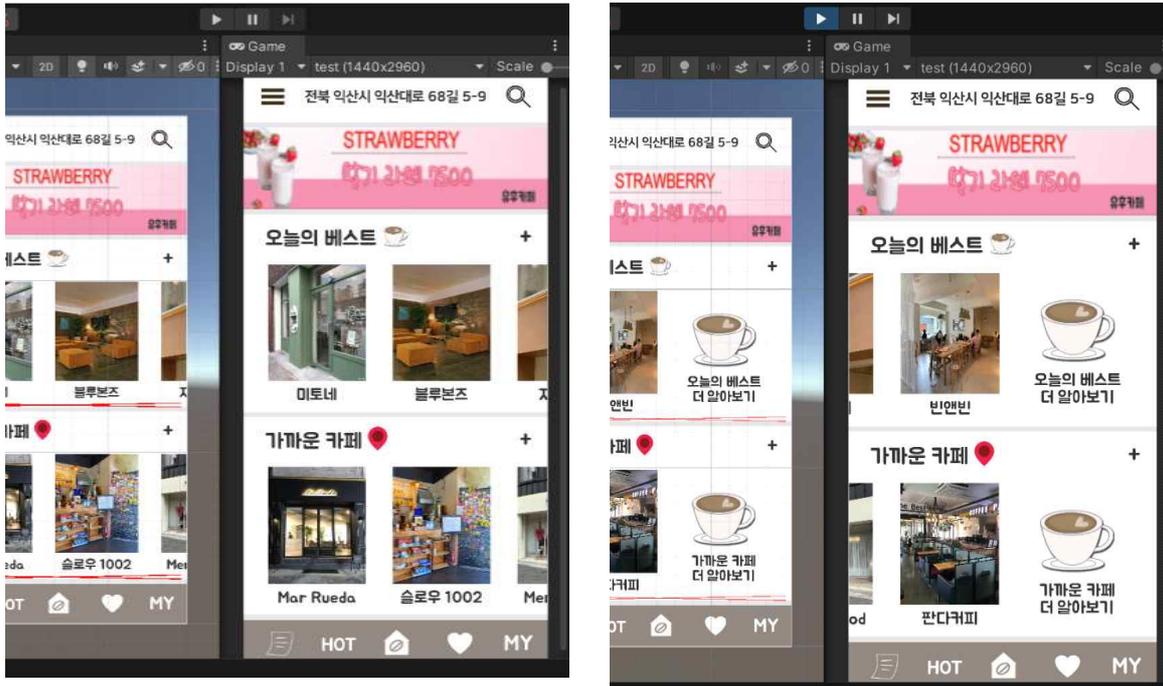


- 프로젝트를 실행하고 File>Build Settings를 선택함



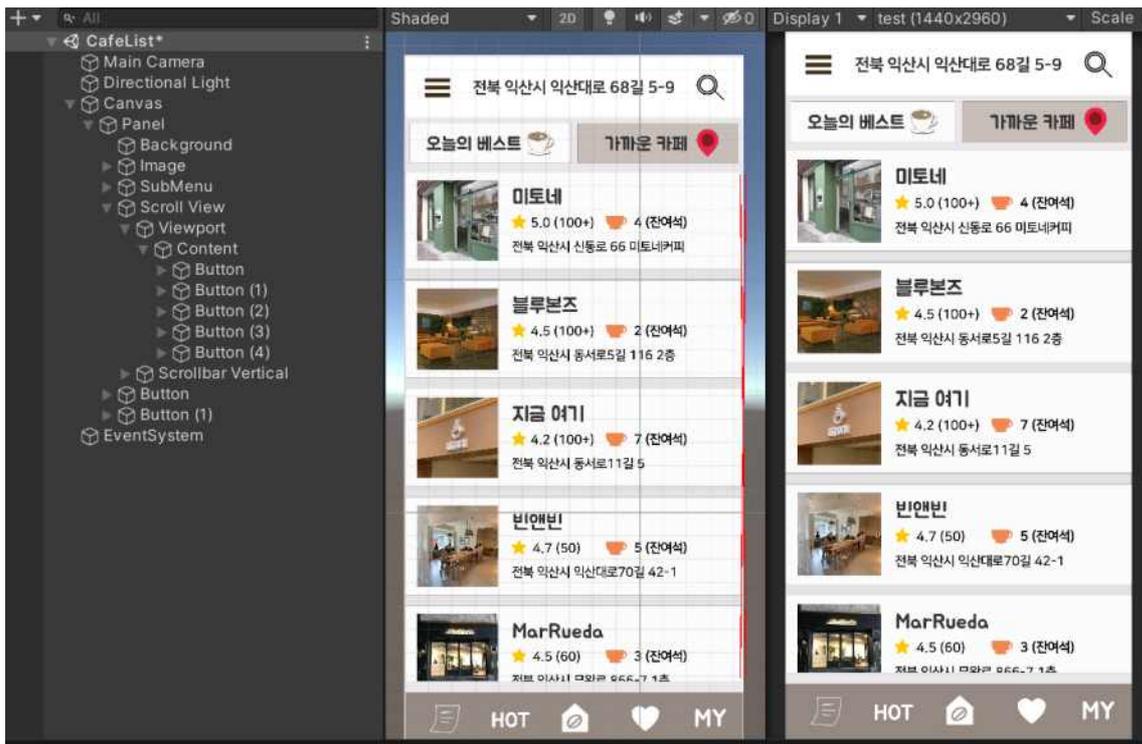
- Android를 선택한 후, Switch Platform을 눌러 설치하고 Build를 해 APK로 내보냄

<CafeIN 메인 화면 구현>



- 왼쪽은 메인 화면이며, 오른쪽은 스크롤 동작 되어 카페의 종류를 확인할 수 있음

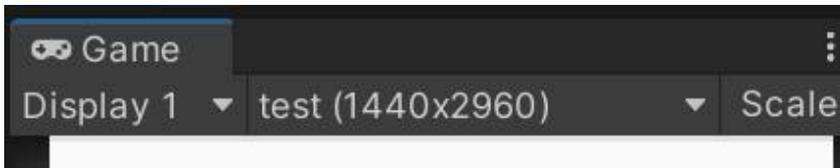
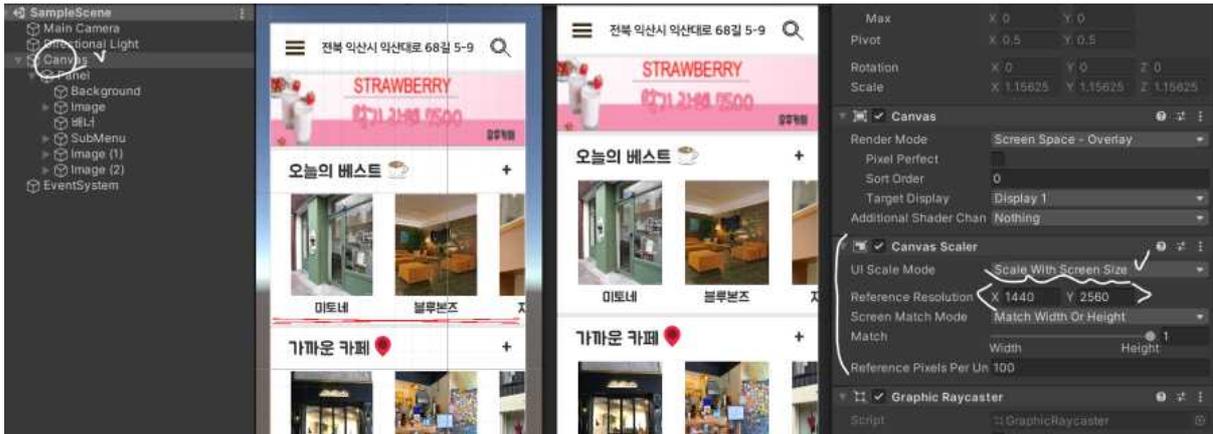
<CafeIN 카페 목록 구현>



<CafeIN 카페 앱 개발 과정>

● 앱 화면 크기 설정

1.Canvas 생성 후 오른쪽과 같이 스마트폰 해상도에 맞춰 크기를 설정한다.

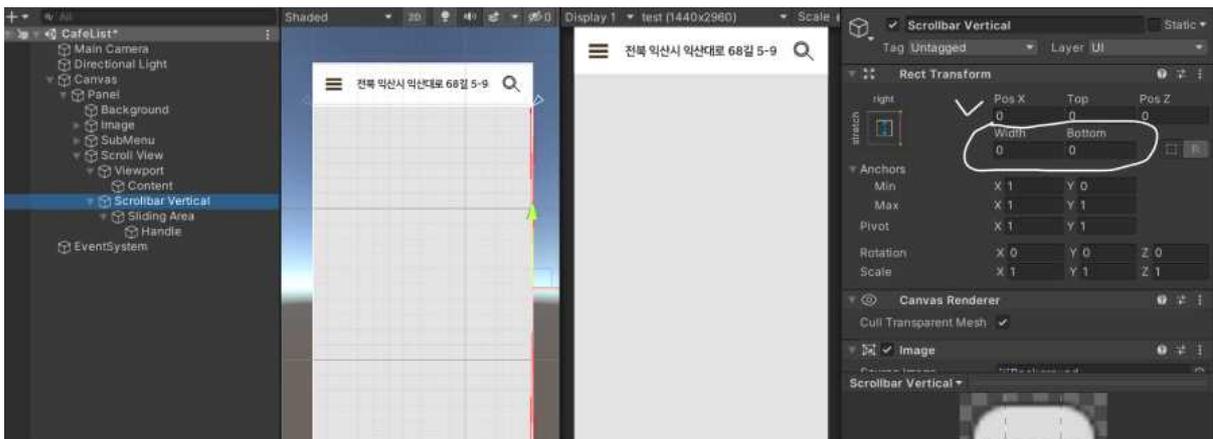


2.Game에서도 Display 크기를 설정해준다.

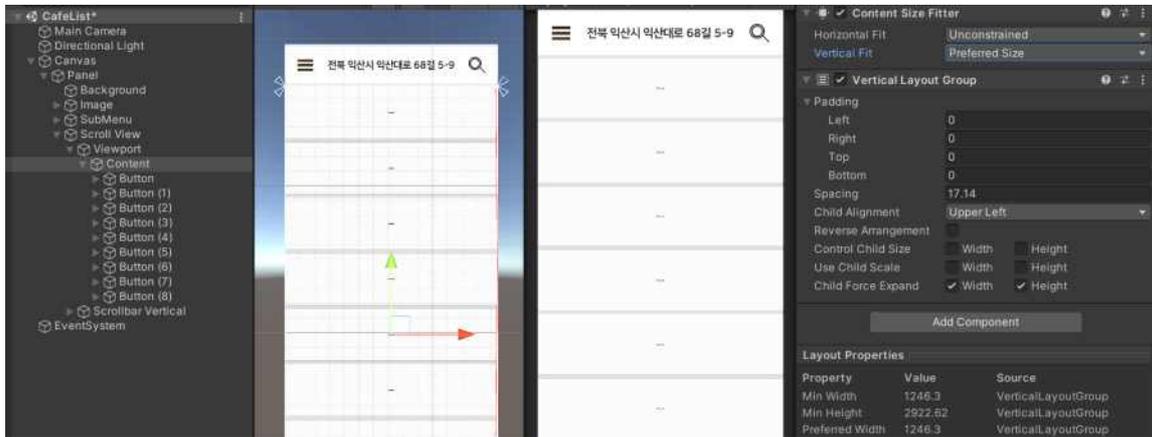
3.Canvas 안에 Panel을 생성해주고 Panel 안에 앱에 들어갈 UI를 추가하여 위치하도록 한다.

● ScrollView에서 UI 구현

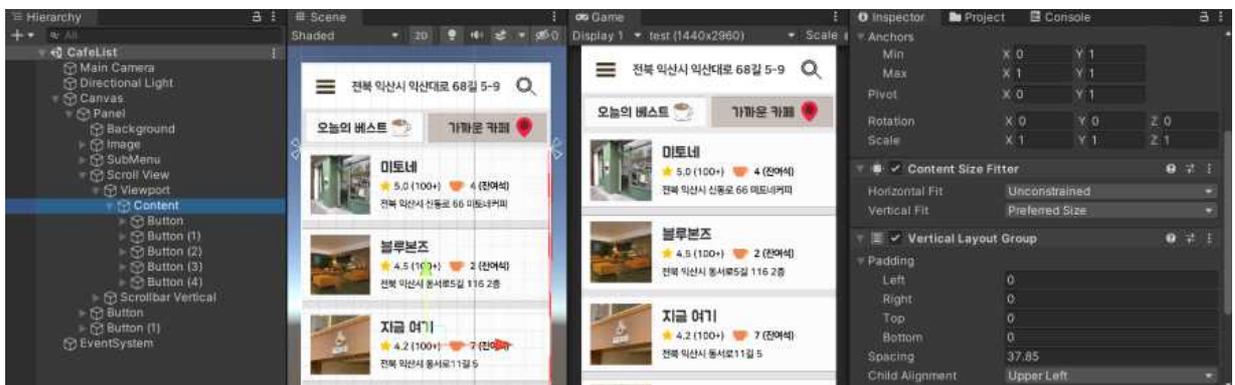
1.Canvas 안에 Scroll View를 추가한 후, 다음 그림에서 왼쪽 부분과 같이 수평인지 수직인지에 따라 Vertical과 Horizontal을 맞게 지워준다.



2. 오른쪽 인스펙터 창에서 Scrollbar Vertical의 Width와 Bottom을 0으로 주어 보이지 않게 설정한다.

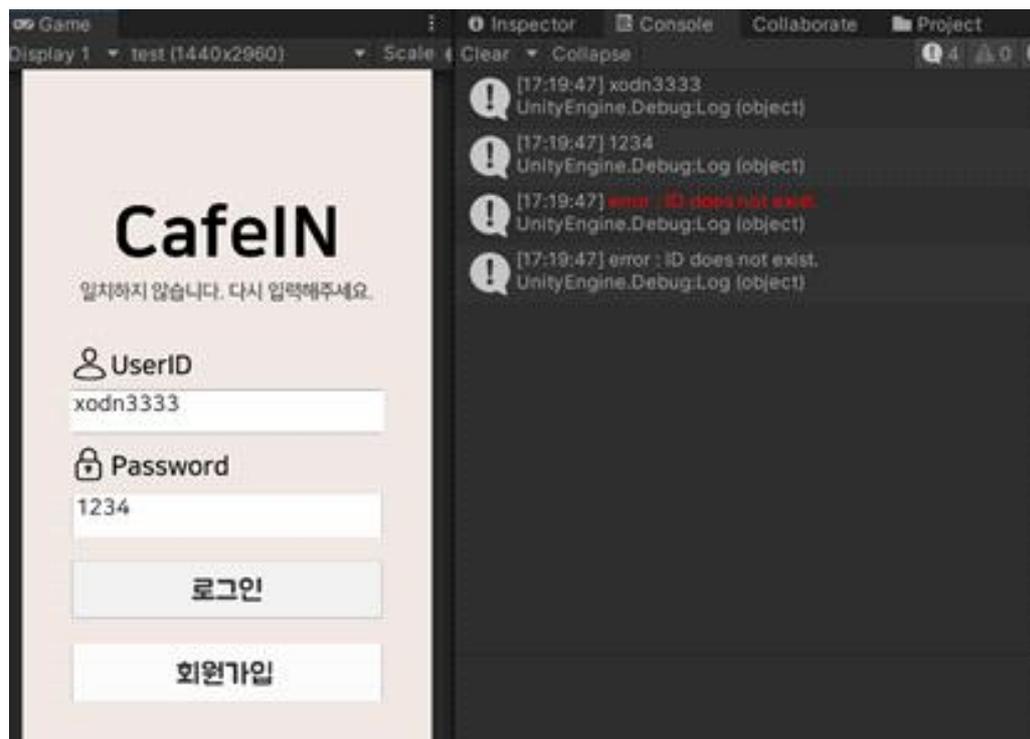


3. 해당 Content 안에 Button을 생성하여 스크롤 기능이 구현되도록 Vertical Layout Group과 Content Size Fitter를 추가하여 동작되도록 한다.

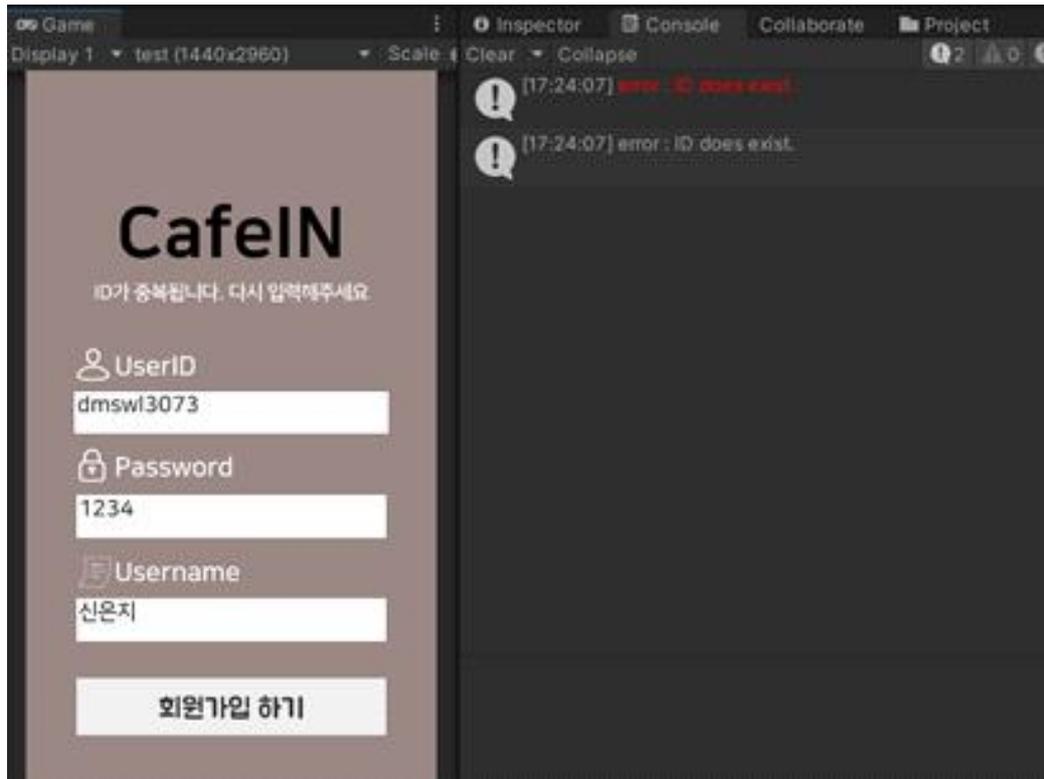


<CafeIN 로그인 동작>

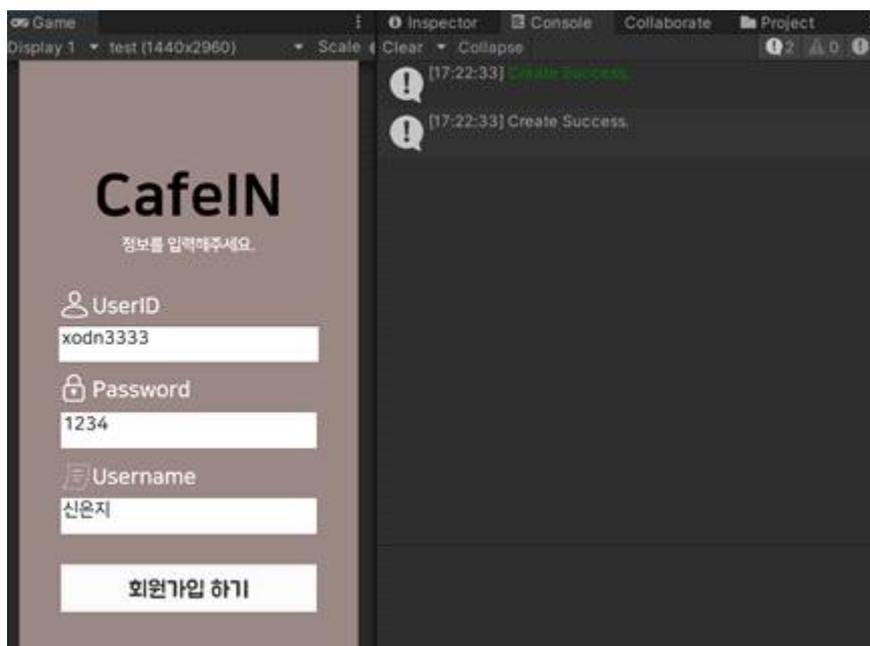
- 회원가입 안 한 ID 로그인 실패

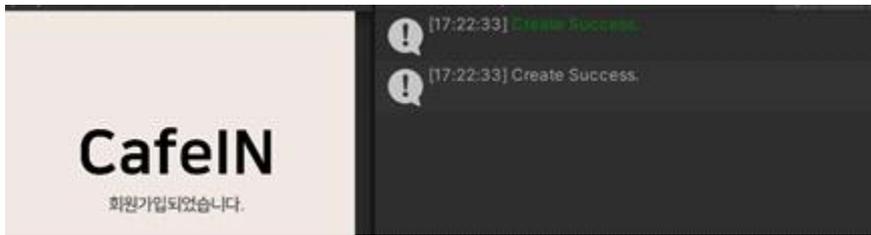


- 회원가입 시, 아이디 중복 -> 회원가입 실패

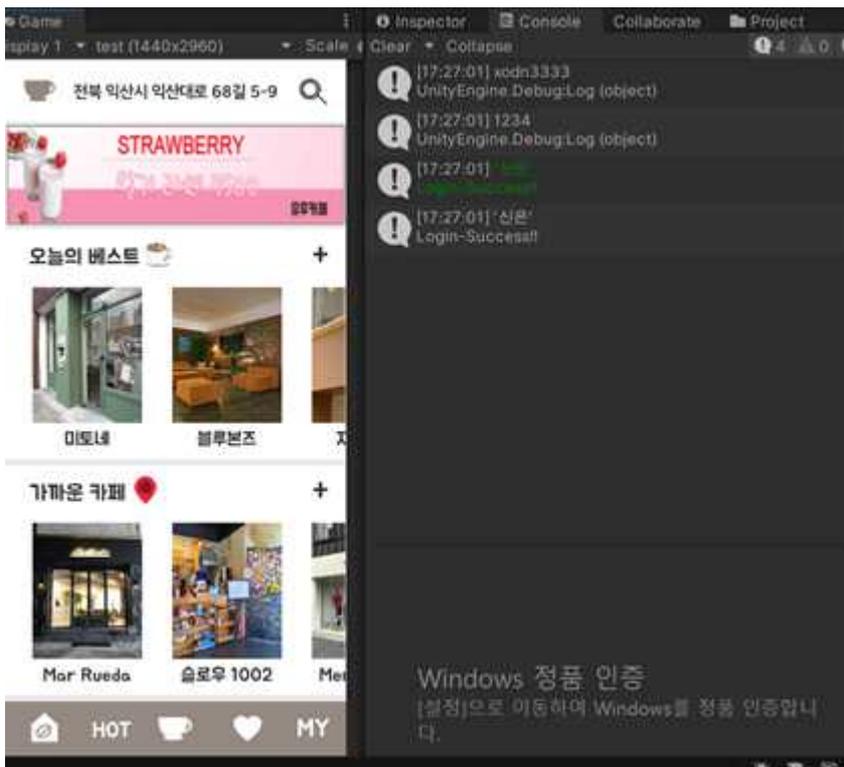


- 회원가입 성공





- 로그인 성공하면 메인 화면으로 전환



- 로그인 UI 배치

InputField와 Text를 생성해 위치시킨다.



GameManager.cs를 작성해 로그인이 동작되도록 한다.

1. phpMyAdmin에서 DB 생성 및 저장

phpMyAdmin 로그인



데이터베이스 생성



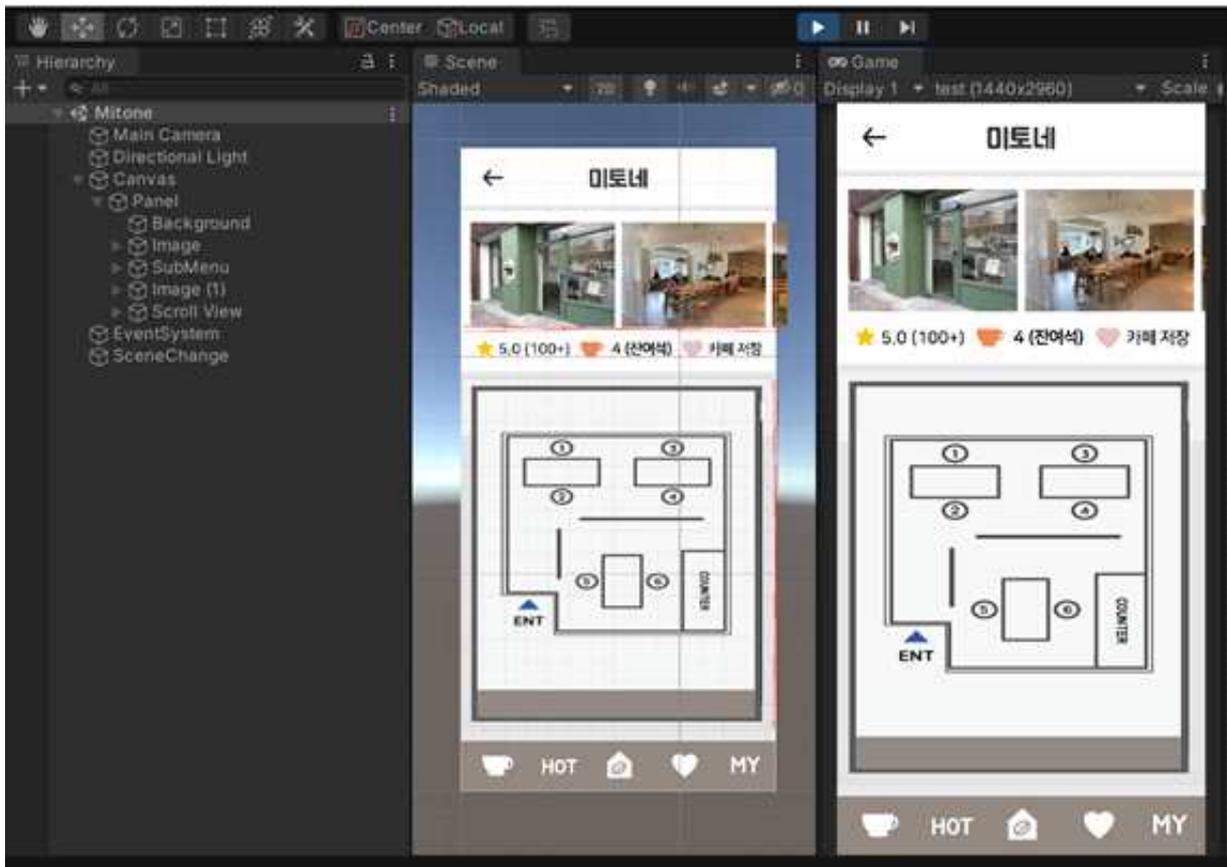
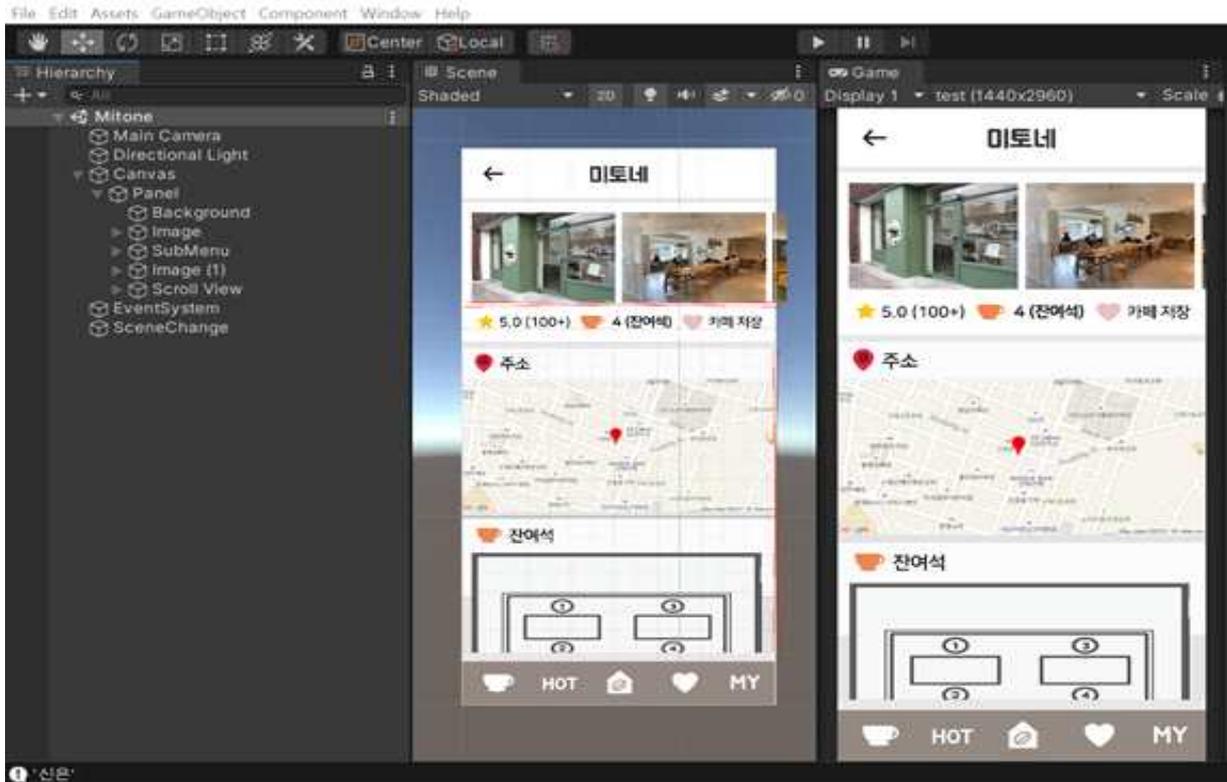
2. .php 파일 작성 후 FileZila에 저장

phpMyAdmin 로그인



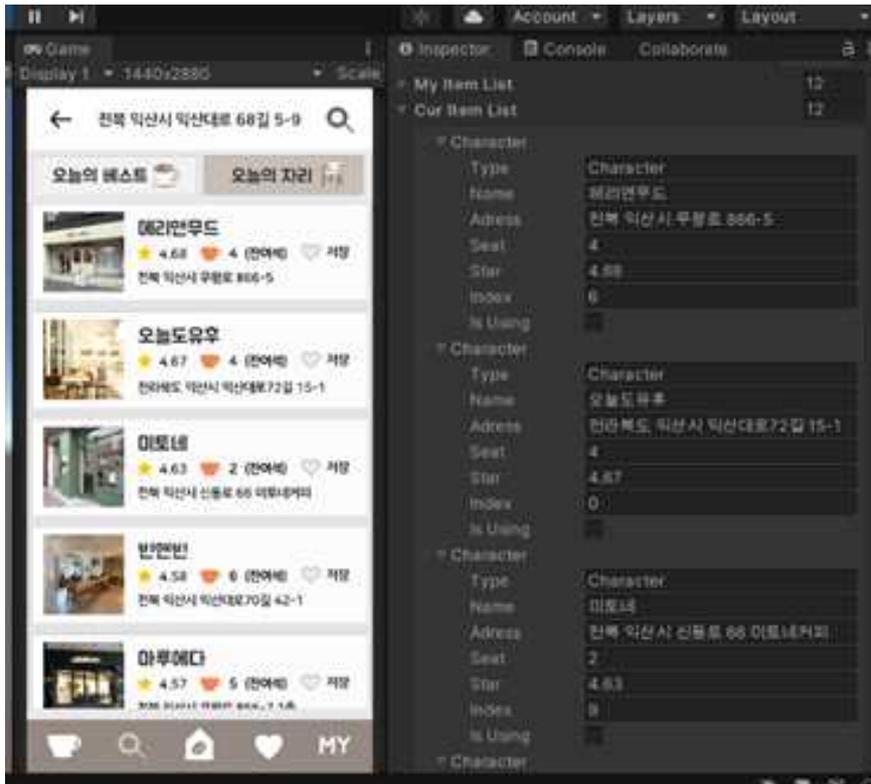
Login.php, CreateAccount.php 파일 작성 하고 실행

<CafeIN 카페 상세 구현>

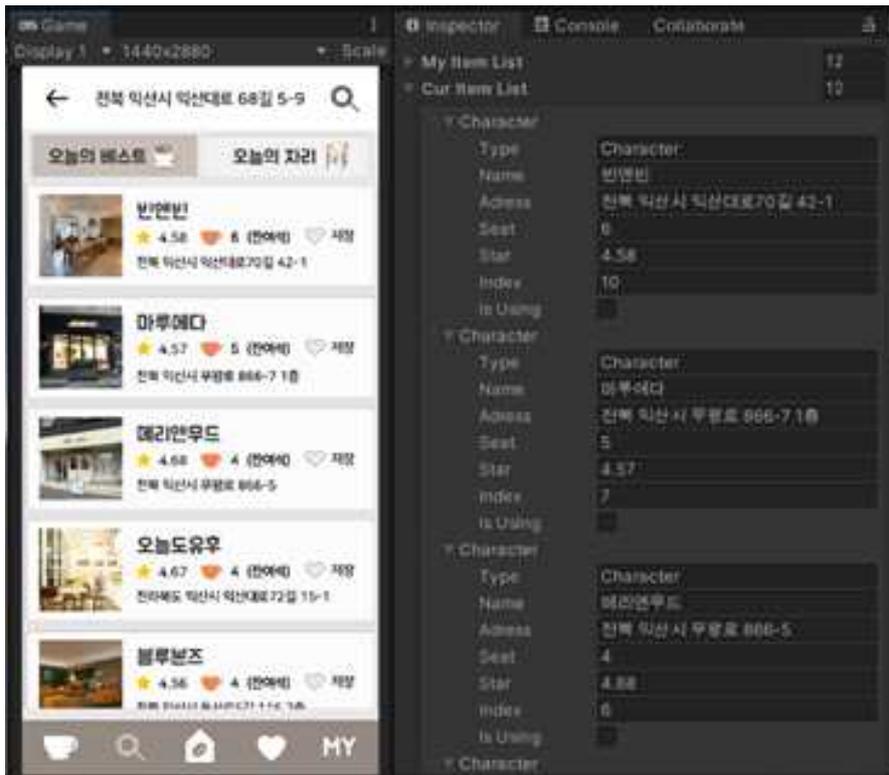


<카페 데이터베이스 구축>

1. 카페 리스트에서 별점 순으로 정렬



2. 카페 리스트에서 좌석순으로 정렬

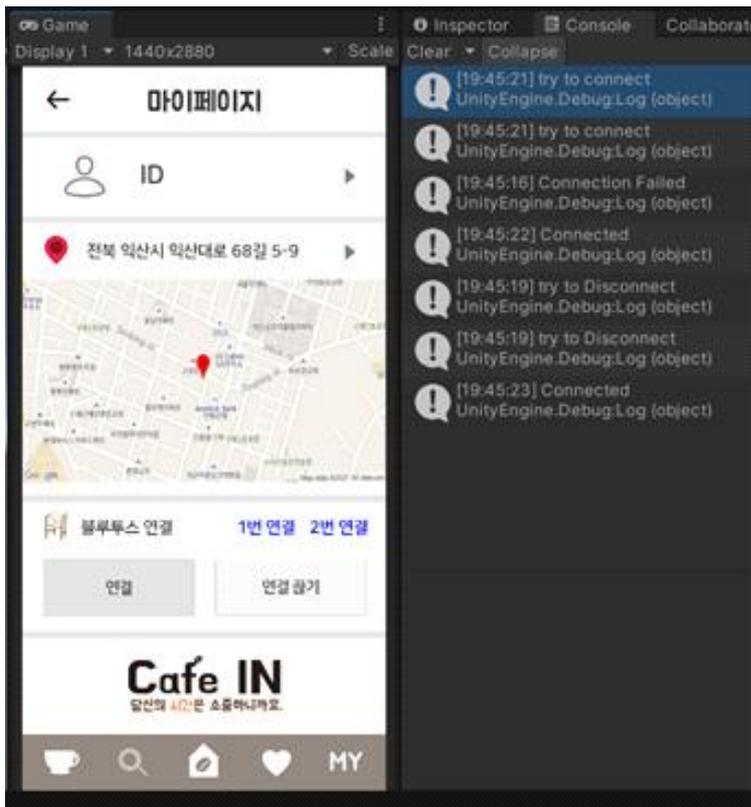


<Owner의 카페 관리 기능>

-Owner는 자기의 카페를 검색해 잔여석을 직접 관리해 데이터를 전송할 수 있음



<카페 좌석 연동 시스템>





- 앱에서 블루투스 연결 버튼을 누르면 해당 벨이 연동됨
- 해당 좌석 벨을 눌렀을 때, 해당 좌석이 켜지고 꺼짐
- 해당 좌석이 켜지면 "사용 중"이어서 잔여석 1개가 줄고, 꺼지면 잔여석 1개가 다시 늘어남

[개발 과정] <카페 데이터베이스 구축>

CafeNarduinno - ItemDatabase.txt

ItemID	Type	Name	Address	Seat	Star	isUsing	Index
1	Character	오늘도유우	전라북도 익산시 익산대로72길 15-1	2	4.67	FALSE	0
2	Character	타임스테이션	전북 익산시 익산대로 460	3	4.46	FALSE	1
3	Character	더리타	전북 익산시 무왕로 876	1	4.43	FALSE	2
4	Character	로제타 커피	전북 익산시 무왕로 870 베스트편입영여	2	4.36	FALSE	3
5	Character	스머틀어와	전북 익산시 익산대로70길 5-35 1층 상가	4	4.35	FALSE	4
6	Character	신동커피	전북 익산시 무왕로 878-13 1층 신동커피	3	4.53	FALSE	5
7	Character	메리앤무드	전북 익산시 무왕로 866-5	4	4.68	FALSE	6
8	Character	마루에다	전북 익산시 무왕로 866-7 1층	5	4.57	FALSE	7
9	Character	슬로우커피	전북 익산시 익산대로70길 8 SLOW 1002	4	4.36	FALSE	8
10	Character	미토네	전북 익산시 신동로 66 미토네커피	2	4.63	FALSE	9
11	Character	빈앤빈	전북 익산시 익산대로70길 42-1	6	4.58	FALSE	10
12	Character	블루부즈	전북 익산시 동서로5길 116 2층	4	4.56	FALSE	11

1.카페에 대한 정보를 데이터베이스로 저장하여 유니티에 연결함

```

GameManagerItem.cs
Assembly-CSharp
using UnityEngine.UI;
using Newtonsoft.Json;
using System.IO;

[System.Serializable]
public class Item {
    public Item(string _Type, string _Name, string _Address, string _Seat, string _Star, bool _isUsing, string _Index)
    {
        Type = _Type; Name = _Name; Address = _Address; Seat = _Seat; Star = _Star; isUsing = _isUsing; Index = _Index;
    }
    public string Type, Name, Address, Seat, Star, Index;
    public bool isUsing;
}
    
```

Unity에서 Json을 사용하여 데이터베이스를 변수로 정보를 저장함

```

GameManagerItem.cs
Assembly-CSharp
GameManagerItem
Load()

void Save() {
    string jsonData = JsonConvert.SerializeObject(MyItemList); //직렬화(한출력 후 나오는) 시점 -> JSON은 String화 함
    File.WriteAllText(Application.dataPath + "/Resources/MyItemText.txt", jsonData); //파일 생성 후 내용 입력 -> 파일에 저장함
}

void Load()
{
    string jsonData = File.ReadAllText(Application.dataPath + "/Resources/MyItemText.txt");
    MyItemList = JsonConvert.DeserializeObject<List<Item>>(jsonData); //string을 List로 직렬화함
}

public void ButtonClick(string tabName)
{
    //현재 아이템 리스트에 클릭한 타입만 추가
    curType = tabName;
    CurItemList = MyItemList.FindAll(x => x.Type == tabName); //가져온 타입에 해당하는 값이면 CurItemList에 대입
    for (int i = 0; i < Slot.Length; i++)
    {
        //슬롯과 텍스트 분하기
        bool isExist = CurItemList.Count;
        Slot[i].SetActive(isExist);
        Slot[i].GetComponentInChildren<Text>().text = isExist ? CurItemList[i].Name : "";
        Slot[i].transform.GetChild(1).GetComponent<Text>().text = isExist ? CurItemList[i].Address : "";
        Slot[i].transform.GetChild(2).GetComponent<Text>().text = isExist ? CurItemList[i].Seat : "";
        Slot[i].transform.GetChild(3).GetComponent<Text>().text = isExist ? CurItemList[i].Star : "";
    }
}
    
```

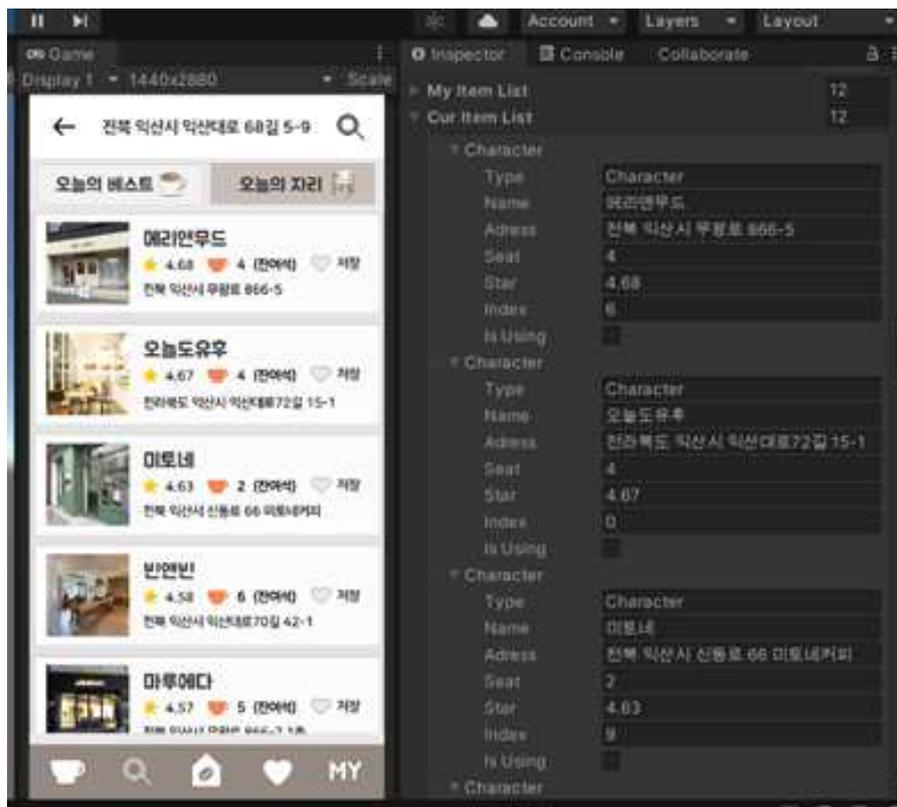
```

GameManagerItem.cs
Assembly-CSharp
GameManagerItem
Load()

202 Slot[i].transform.GetChild(3).GetComponent<Text>().text = isExist ? CurItem[i].Star : "";
203
204 //아이템 이미지와 사용중인지 보이기
205 if (isExist)
206 {
207     ItemImage[i].sprite = ItemSprite[AllItemList.FindIndex(x => x.Name == CurItem[i].Name)];
208     UsingImage[i].SetActive(CurItem[i].isUsing);
209 }
210 }
211 }
212 }
213 }
214
215 public void TabClick(string tabName)
216 {
217     MyItemList.Sort((p1, p2) => p2.Star.CompareTo(p1.Star));
218     //현재 아이템 리스트에 클릭한 타입만 추가
219     curType = tabName;
220     CurItemList = MyItemList.FindAll(x => x.Type == tabName); //O/X의 타입이 일치하면 CurItemList에 대입
221
222     for (int i = 0; i < Slot.Length; i++)
223     {
224         //슬롯과 텍스트 보이기
225         bool isExist = i < CurItemList.Count;
226         Slot[i].SetActive(isExist);
227         Slot[i].GetComponentInChildren<Text>().text = isExist ? CurItemList[i].Name : "";
228         Slot[i].transform.GetChild(1).GetComponent<Text>().text = isExist ? CurItemList[i].Address : "";
229         Slot[i].transform.GetChild(2).GetComponent<Text>().text = isExist ? CurItemList[i].Seat : "";
230         Slot[i].transform.GetChild(3).GetComponent<Text>().text = isExist ? CurItemList[i].Star : "";
231
232         //아이템 이미지와 사용중인지 보이기
233         if (isExist) {
234             ItemImage[i].sprite = ItemSprite[AllItemList.FindIndex(x => x.Name == CurItem[i].Name)];
235             UsingImage[i].SetActive(CurItem[i].isUsing);
236         }
237     }
238 }

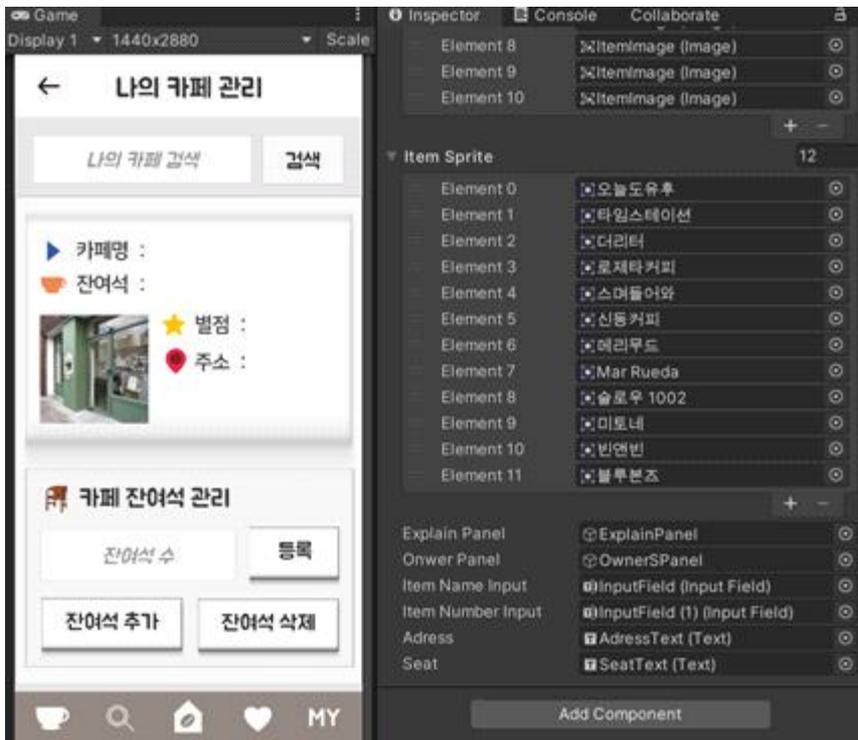
```

저장한 변수를 불러와 데이터베이스를 연동시키는 스크립트를 통해 Unity UI와 연결되도록 작성함



해당 스크립트를 연결하여 Inspector에서 변수가 자동으로 데이터를 받는 것을 확인할 수 있음

<Owner의 카페 관리 기능>



GameManagerItem에서 검색 Field와 Button UI를 연결해 데이터가 저장되고 생성되도록 함

```

GameManagerItem.cs
Assembly-CSharp
GameManagerItem
GetItem5

53 public void GetItemSearch()
54 {
55     Item curItem = MyItemList.Find(x => x.Name == ItemNameInput.text); // 아이템 검색
56     int i = int.Parse(curItem.Index);
57     //ItemImage[i].sprite = ItemSprite[AllItemList.FindIndex(x => x.Name == CurItemList[i].Name)];
58
59     OwnerPanel.transform.GetChild(2).GetComponentInChildren<Text>().text = curItem.Name;
60     //OwnerPanel.transform.GetChild(4).GetComponent<Image>().sprite = Slot[slotNum].transform.GetChild
61     OwnerPanel.transform.GetChild(3).GetComponent<Text>().text = curItem.Address;
62     OwnerPanel.transform.GetChild(4).GetComponent<Text>().text = curItem.Seat;
63     OwnerPanel.transform.GetChild(5).GetComponent<Text>().text = curItem.Star;
64     OwnerPanel.transform.GetChild(6).GetComponent<Image>().sprite = ItemSprite[i];
65
66 }
67 //조작
68 public void GetItemClick() {
69     Item curItem = MyItemList.Find(x => x.Name == ItemNameInput.text); // 아이템 검색
70     ItemNumberInput.text = ItemNumberInput.text == "" ? "1" : ItemNumberInput.text;
71     if (curItem != null)
72     { //숫자 조정
73         curItem.Seat = (int.Parse(curItem.Seat) + int.Parse(ItemNumberInput.text)).ToString();
74     }
75     else {
76         //전체에서 얻을 아이템을 찾아 내 아이템을 추가
77         Item curAllItem = AllItemList.Find(x => x.Name == ItemNameInput.text);
78         if (curAllItem != null)
79         {
80             curAllItem.Seat = ItemNumberInput.text;
81             MyItemList.Add(curAllItem);
82         }
83     }
84     MyItemList.Sort((p1, p2) => p2.Seat.CompareTo(p1.Seat));
85     Save();

```

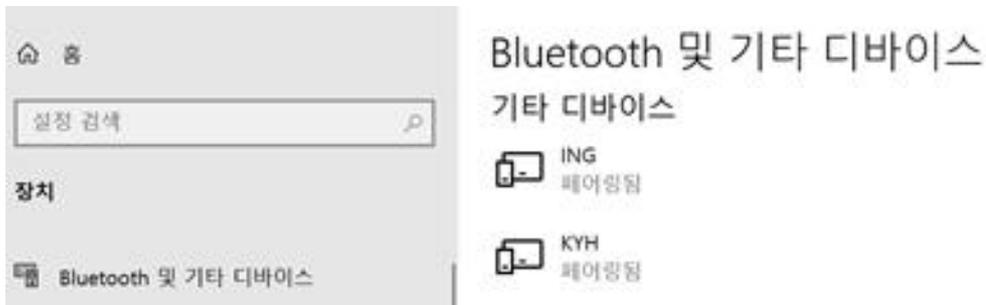
```

GameManagerItem.cs
Assembly-CSharp GameManagerItem
87 public void RemoveItemClick()
88 {
89     Item curItem = MyItemList.Find(x => x.Name == itemNameInput.text);
90     if (curItem != null) {
91         int curNumber = int.Parse(curItem.Seat) - int.Parse(itemNumberInput.text);
92
93         if (curNumber <= 0) MyItemList.Remove(curItem);
94         else curItem.Seat = curNumber.ToString();
95     }
96
97     MyItemList.Sort((p1, p2) => p1.Seat.CompareTo(p2.Seat));
98     Save();
99 }
100

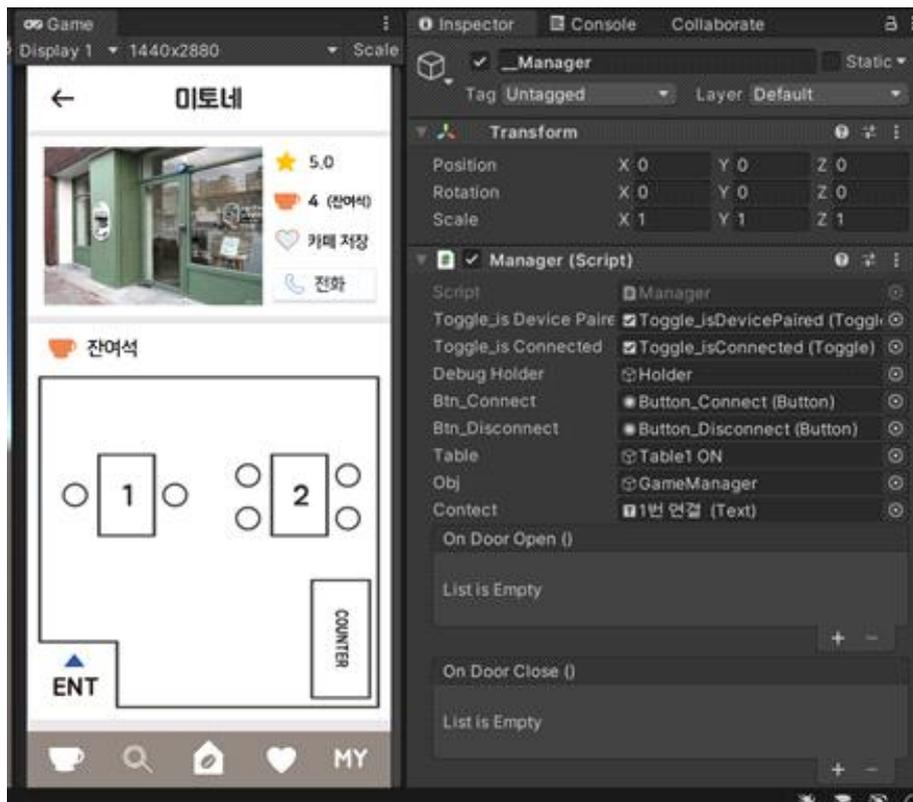
```

스크립트에서 InputField에서 받은 잔여석 수가 추가되고 제거되도록 연산자에 대해 코드를 작성함

<카페 좌석 연동 시스템>



Manager 스크립트를 통해 아두이노의 블루투스가 연동되도록 하고, 해당 함수가 동작되도록 Canvas UI와 연결함



```

CafeHerdino - Manager.cs
Manager.cs
Assembly-CSharp Manager bluetoothHelper
33 // 왼쪽에서 접근하여 사용될 수 있습니다
34 // 외부 다른 클래스에서 Manager 함수로 접근하여 Start 함수에서 다음과 같이 사용하시면 됩니다
35 //
36 // Manager.onDoorOpen.AddListener(OnDoorOpen);
37 // Manager.onDoorClose.AddListener(OnDoorClose);
38 //
39 public UnityEvent onDoorOpen, onDoorClose;
40
41 string Red = "<color=FF0000>" + "1번 오류" + "</color>";
42 string Blue = "<color=0000FF>" + "1번 연결" + "</color>";
43
44 #Unity 메시지가 잘못 오면
45 void Start () {
46
47     Btn_Connect.onClick.AddListener (() => {
48         //ConnectFast.SetActive(true);
49         if (bluetoothHelper.isDevicePaired ()) {
50             Debug.Log ("try to connect");
51             Connect.gameObject.SetActive(true);
52             Connect.text = ("연결 중.");
53             bluetoothHelper.Connect (); // tries to connect
54         } else {
55             Debug.Log ("not DevicePaired");
56             Connect.text = Red;
57         }
58     });
59     Btn_Disconnect.onClick.AddListener (() => {
60         bluetoothHelper.Disconnect ();
61         Debug.Log ("try to Disconnect");
62     });
63     //=====
64     //=====
65     deviceName = "KIM"; //bluetooth should be turned ON. // 제어당되는 아두이노 블루투스 이름과 같아야 합니다.
66     //=====
67     //=====
68     try {
69         bluetoothHelper = BluetoothHelper.GetInstance (deviceName);
70
71     try {
72         bluetoothHelper = BluetoothHelper.GetInstance (deviceName);
73
74     try {
75         bluetoothHelper = BluetoothHelper.GetInstance (deviceName);
76         bluetoothHelper.OnConnected += OnConnected;
77         bluetoothHelper.OnConnectionFailed += OnConnectionFailed;
78         bluetoothHelper.OnDataReceived += OnMessageReceived; //read the data
79
80         bluetoothHelper.setTerminatorBasedStream ("*");
81
82         if (bluetoothHelper.isDevicePaired ())
83             Toggle_isDevicePaired.isOn = true;
84         else
85             Toggle_isDevicePaired.isOn = false;
86     } catch (Exception ex) {
87         Toggle_isDevicePaired.isOn = false;
88         Debug.Log (ex.Message);
89     }
90
91 // Update is called once per frame
92 #Unity 메시지가 잘못 오면
93 void Update () {
94
95     if (Input.GetKeyDown (KeyCode.Alpha0)) {
96         if (!isDebugOn) {
97             isDebugOn = true;
98             DebugHolder.SetActive (true);
99         } else {
100             isDebugOn = false;
101             DebugHolder.SetActive (false);
102             myLog = "reset";
103         }
104     }
105
}

```

블루투스를 페어링한 후, 스크립트에 해당 블루투스의 이름을 작성함.
이름에 해당하는 블루투스의 연결을 받아와 아두이노와 연동되도록 함

```

Manager.cs
Assembly-CSharp
Manager
bluetooth

103 //Asynchronous method to receive messages
104 참조 1개
105 void OnMessageReceived () {
106     received_message = bluetoothHelper.Read ();
107     Debug.Log(received_message);
108
109     if (received_message.Contains ("on")) {
110         //Txt_Door.text = "Door is close";
111         Debug.Log("1번 자리 있음");
112         table.SetActive(true);
113         obj.GetComponent<GameManagerItem>().RemoveSeat();
114         onDoorClose.Invoke();
115     }
116
117     if (received_message.Contains ("off")) {
118         //Txt_Door.text = "Door is open";
119         Debug.Log("1번 자리 없음");
120         table.SetActive(false);
121         obj.GetComponent<GameManagerItem>().GetSeat();
122         onDoorOpen.Invoke();
123     }
124 }
125
126 참조 1개
127 void OnConnected () {
128     Toggle_isConnected.isOn = true;
129     try {
130         bluetoothHelper.StartListening ();
131         Debug.Log ("Connected");
132
133         Connect.text = Blue;
134     } catch (Exception ex) {
135         Debug.Log (ex.Message);
136     }
137 }

```

아두이노에서 터치 센서의 ON,OFF를 받아들 때 해당 좌석의 Image가 켜지고 꺼지는 것을 스크립트로 작성해 연결함
 블루투스 오류일 시, Text가 빨간색으로 보여지고 블루투스가 연결되면 Text가 파란색으로 보여지도록 작성한다.

<앱 시작 화면 애니메이션>

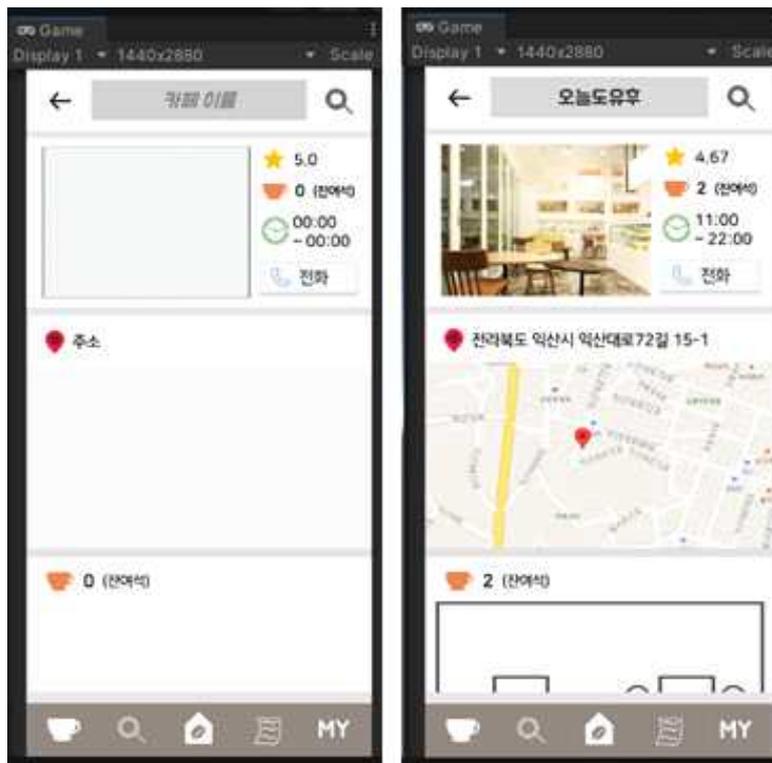


<카페마다 좌석 배치도 연결>



: 해당하는 카페의 좌석 배치도를 UI로 배치해 제공한다.

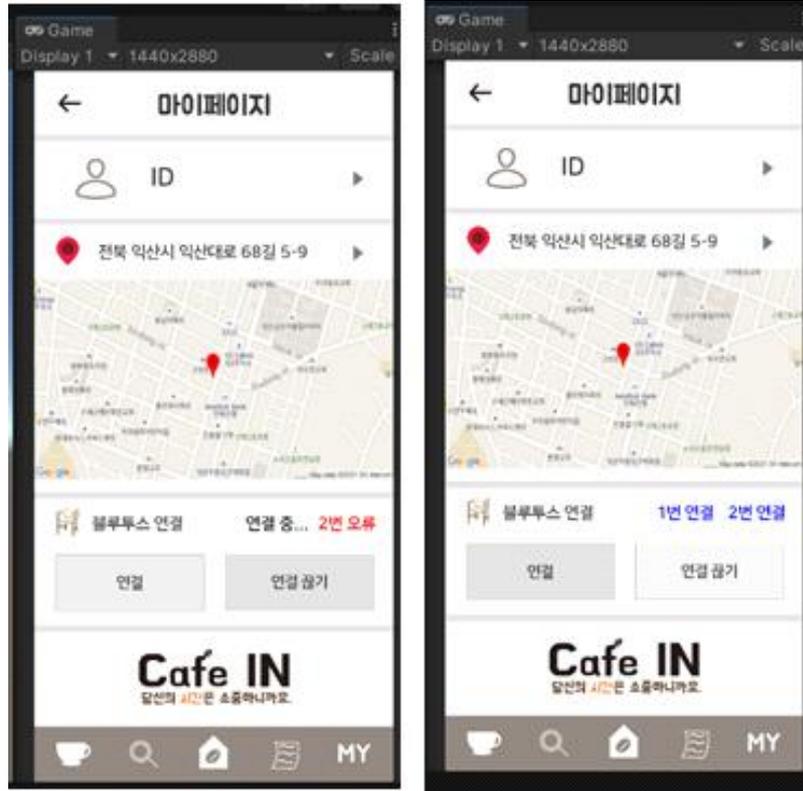
<카페 검색 기능>



: 원하는 카페를 검색하면 정보를 얻을 수 있다.

<블루투스 연결 여부 확인>

: 마이페이지에서 Owner는 좌석벨과의 블루투스 연결 여부를 확인할 수 있다.



[개발 과정]

< 앱 시작 화면 애니메이션 설정 >



1. 각각 Sprite 이미지에 대한 투명도의 값을 주어 애니메이션을 만든다.



2. 그리고 애니메이션 후 바로 앱의 로그인 창이 뜨도록 LoadScene을 해준다.

```

CafeINArduino - SceneChange.cs
SceneChange.cs
Assembly-CSharp
SceneChange

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  @Unity 스크립트 | 참조 0개
7  public class SceneChange : MonoBehaviour
8  {
9      @Unity 메시지 | 참조 0개
10     void Start()
11     {
12         Invoke("sceneChange", 4);
13     }
14     참조 0개
15     void sceneChange()
16     {
17         SceneManager.LoadScene("CafeIN");
18     }
19 }
    
```

< 카페 좌석 배치도 제공 기능>



1.카페의 좌석 배치도를 UI로 만들어 Panel로 저장한 후, GameManager 인스펙터 창에 연결한다.

```

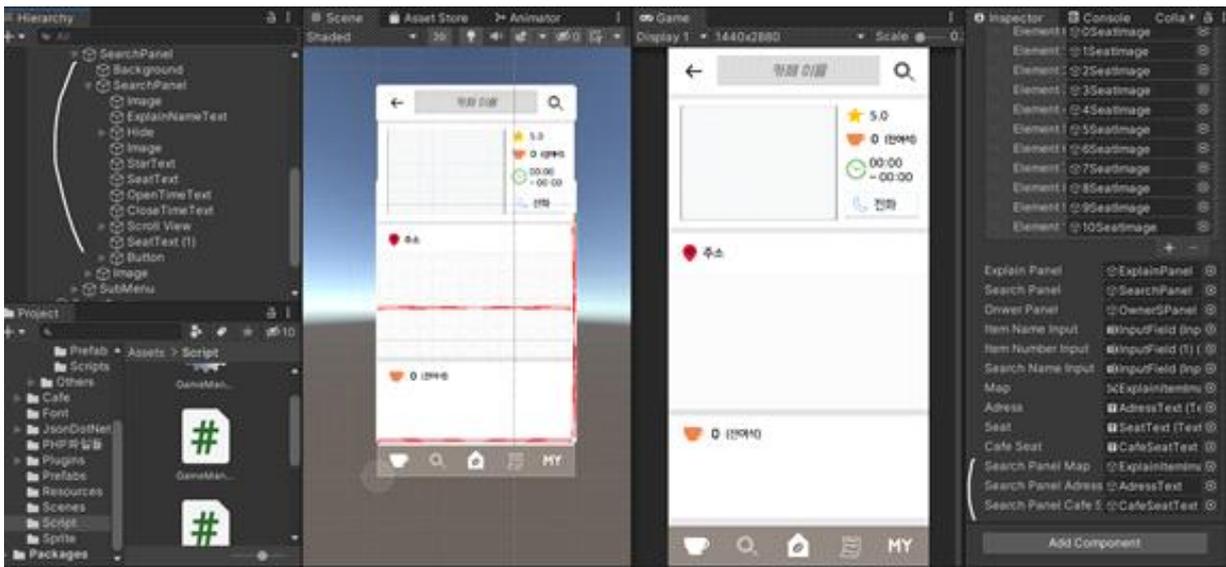
GameManagerItem.cs
Assembly-CSharp
Item
Item(string_Type, string_Name, st

178 public void PointerClick(int slotNum) {
179     print(slotNum + "클릭");
180     ExplainPanel.SetActive(true);
181     Item curItem = MyItemList[slotNum]; // 아이템 검색
182     int i = int.Parse(curItem.Index);
183     for (int a = 0; a < 11; a++)
184     {
185         SeatList[a].SetActive(false);
186     }
187     //새로운 상태
188     ExplainPanel.transform.GetChild(1).GetComponent<Text>().text = CurItemList[slotNum].Name;
189     ExplainPanel.transform.GetChild(3).GetComponent<Image>().sprite = Slot[slotNum].transform.GetChild(4).GetComponent<Image>().sprite;
190     ExplainPanel.transform.GetChild(4).GetComponent<Text>().text = CurItemList[slotNum].Star;
191     ExplainPanel.transform.GetChild(5).GetComponent<Text>().text = CurItemList[slotNum].Seat;
192     ExplainPanel.transform.GetChild(6).GetComponent<Text>().text = AllItemList[i].Open;
193     ExplainPanel.transform.GetChild(7).GetComponent<Text>().text = AllItemList[i].Close;
194     Map.GetComponent<Image>().sprite = MapSprite[i];
195     SeatList[i].SetActive(true);
196     // SeatList[slotNum].SetActive(false);
197     //GameObject SeatMap = Instantiate(SeatList[i]) as GameObject;
198     Adress.GetComponent<Text>().text = CurItemList[slotNum].Adress;
199     CafeSeat.GetComponent<Text>().text = CurItemList[slotNum].Seat;
200
201 }
202
203 public void CallButton(int slotNum) {
204     Item curItem = MyItemList[slotNum]; // 아이템 검색
205     int i = int.Parse(curItem.Index);
206     Application.OpenURL("tel:" + AllItemList[i].Phone);
207     print("tel:" + AllItemList[i].Phone);
208 }

```

2. GameManagerItem 스크립트에 SeatPanel 리스트를 선언한 후, 해당 카페에 맞게 패널이 뜨도록 코드를 작성한다.

<카페 검색 기능>



1. Search 버튼을 누르면 Search Panel이 나타나도록 GameManager 인스펙터 창에 연결하고 각 데이터가 연결되도록 스크립트로 설정한다.

```

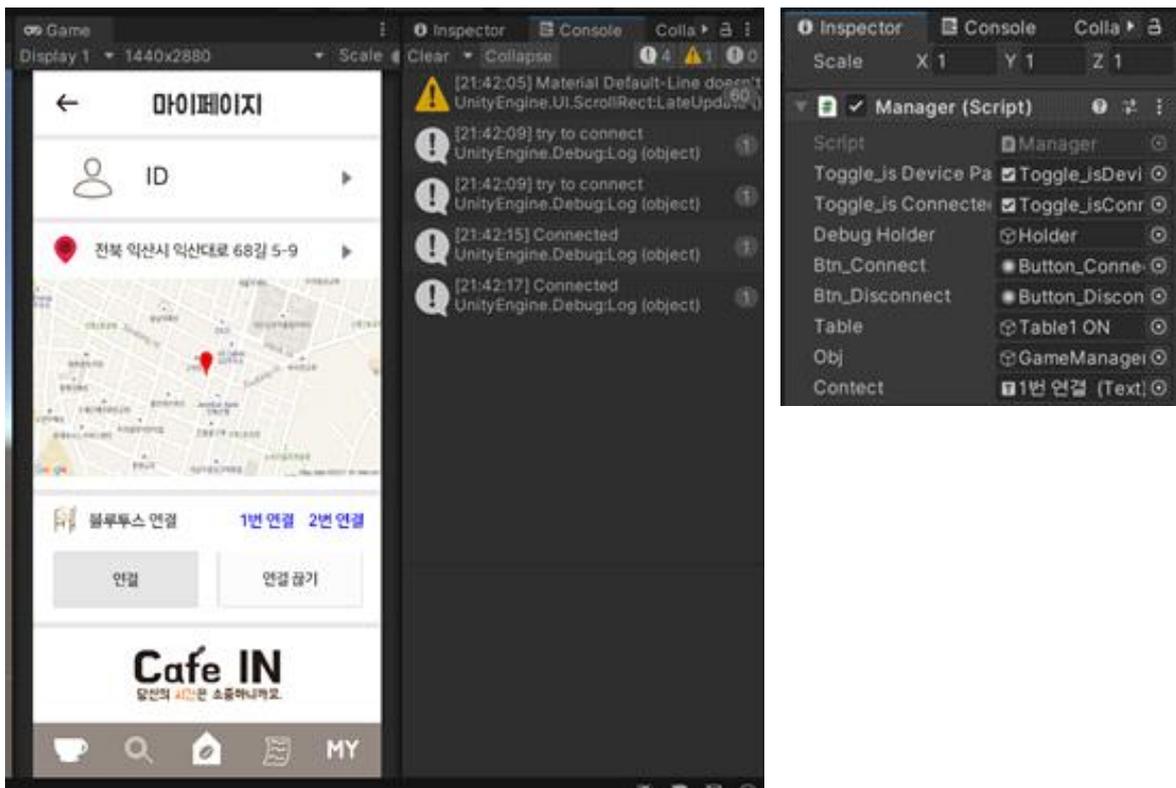
GameManagerItem.cs
Assembly-CSharp
Item
Item

50 public void ItemSearch()
51 {
52     Item curItem = MyItemList.Find(x => x.Name == SearchNameInput.text); // 아이템 검색
53     int i = int.Parse(curItem.Index);
54     //ItemImage[i].sprite = ItemSprite[AllItemList.FindIndex(x => x.Name == CurItemList[i].Name)];
55     for (int a = 0; a < 11; a++)
56     {
57         SeatList[a].SetActive(false);
58     }
59
60     SearchPanel.transform.GetChild(1).GetComponentInChildren<Text>().text = curItem.Name;
61     SearchPanel.transform.GetChild(3).GetComponent<Image>().sprite = ItemSprite[i];
62     SearchPanel.transform.GetChild(4).GetComponent<Text>().text = curItem.Star;
63     SearchPanel.transform.GetChild(5).GetComponent<Text>().text = curItem.Seat;
64     SearchPanel.transform.GetChild(6).GetComponent<Text>().text = AllItemList[i].Open;
65     SearchPanel.transform.GetChild(7).GetComponent<Text>().text = AllItemList[i].Close;
66     SearchPanelMap.GetComponent<Image>().sprite = MapSprite[i];
67     SearchSeatList[i].SetActive(true);
68     // SeatList[slotNum].SetActive(false);
69     //GameObject SeatMap = Instantiate(SeatList[i]) as GameObject;
70     SearchPanelAddress.GetComponent<Text>().text = curItem.Address;
71     SearchPanelCafeSeat.GetComponent<Text>().text = curItem.Seat;
72
73 }

```

2. SeatList에 해당하는 인덱스번호들은 처음에 안보이도록 false값을 주고, 해당하는 인덱스의 SeatList는 보이도록 true값을 준다.

< 블루투스 연결 여부 확인 기능 >



1.블루투스가 연결되면 UI로 띄워지도록 Manager 스크립트에 작성해 연결시킨다.

```

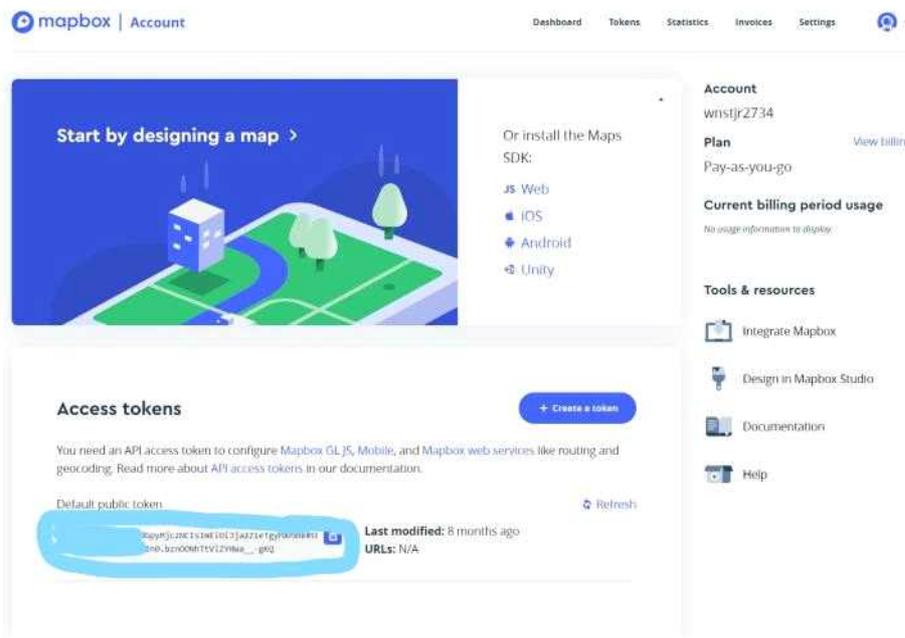
Manager.cs
Assembly-CSharp
Manager
bluetoothHelper

39 public UnityEvent onDoorOpen, onDoorClose;
40
41 string Red = "<color=#FF0000>" + "1번 오류" + "</color>";
42 string Blue = "<color=#0000FF>" + "1번 연결" + "</color>";
43
44 @Unity 메시지 참조 0개
45 void Start () {
46
47     Btn_Connect.onClick.AddListener (() => {
48         //ConnectText.SetActive(true);
49         if (bluetoothHelper.isDevicePaired ()) {
50             Debug.Log ("try to connect");
51             Connect_gameObject.SetActive(true);
52             Connect.text = ("연결 중...");
53             bluetoothHelper.Connect (); // tries to connect
54         } else {
55             Debug.Log ("not DevicePaired");
56             Connect.text = Red;
57         }
58     });
59     Btn_Disconnect.onClick.AddListener (() => {
60         bluetoothHelper.Disconnect ();
61         Debug.Log ("try to Disconnect");
62     });

```

3) GPS 기반의 카페 위치 지도 맵

1. <https://account.mapbox.com>(맵박스 사이트)에서 유니티에서 맵을 띄울 수 있는 API 액세스 토큰을 받는다.



2. 맵박스의 액세스 토큰을 가져오게 되면 승인이 맵박스 승인이 되면서 여러 맵들을 사용 할 수 있는데 그 중 zoomable map을 사용

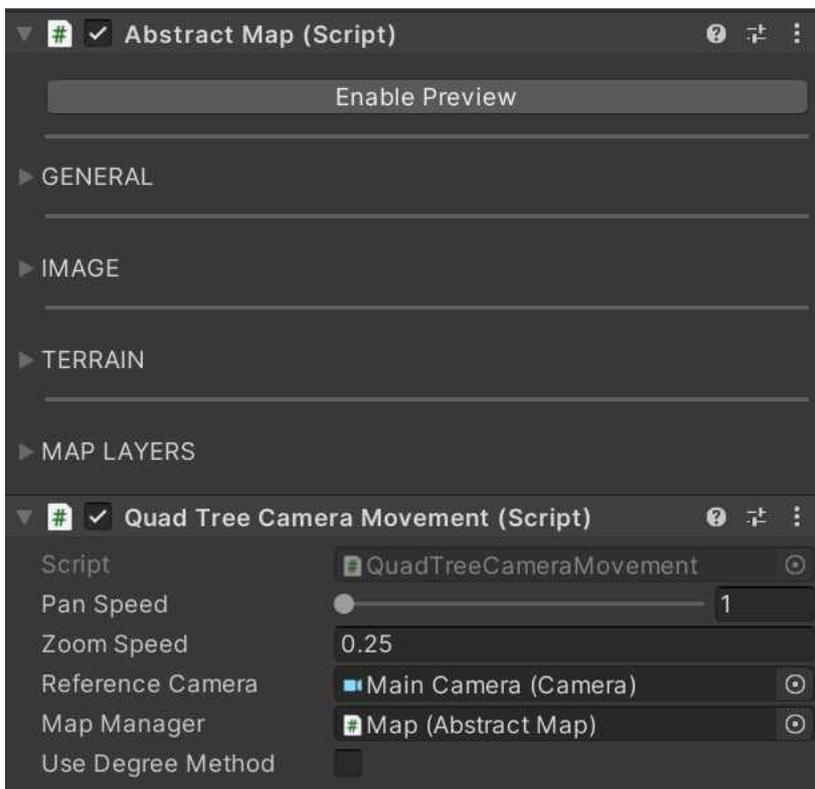


3. zoomableMap 씬을 Import 파일로 제작 후 Map오브젝트를 프리팹으로 제작

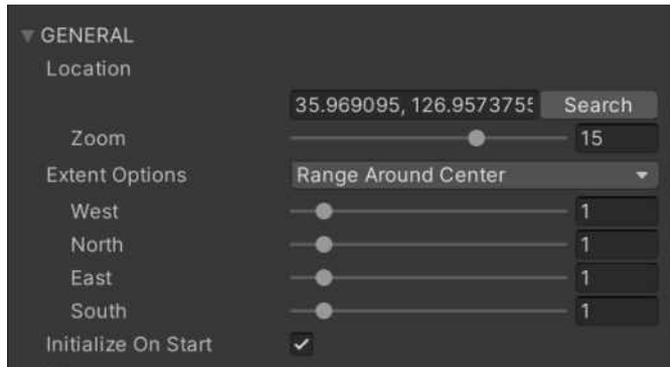


4. map 오브젝트 스크립트 컴포넌트 조정

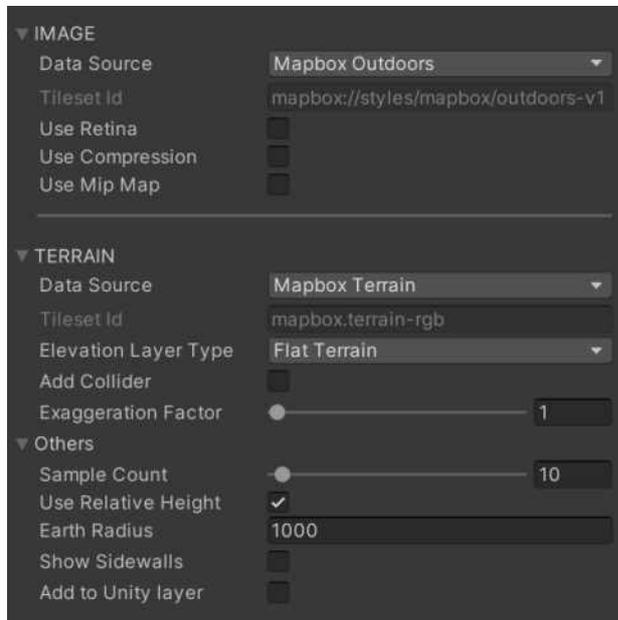
- Abstract Map 스크립트/Quad Tree Camera Movement 스크립트



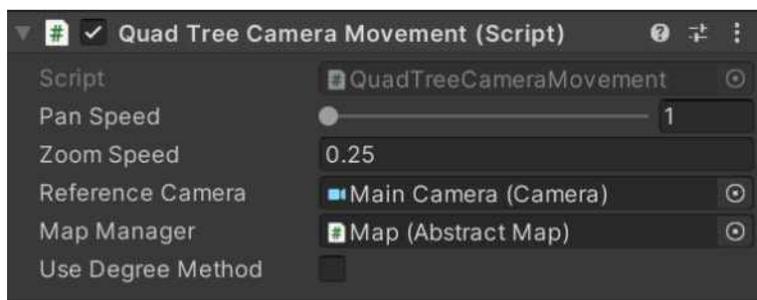
- Abstract Map의 General 옵션
 - Location(위도, 경도 설정), Zoom(맵의 확대,축소)
 - Location을 프라임관으로 설정



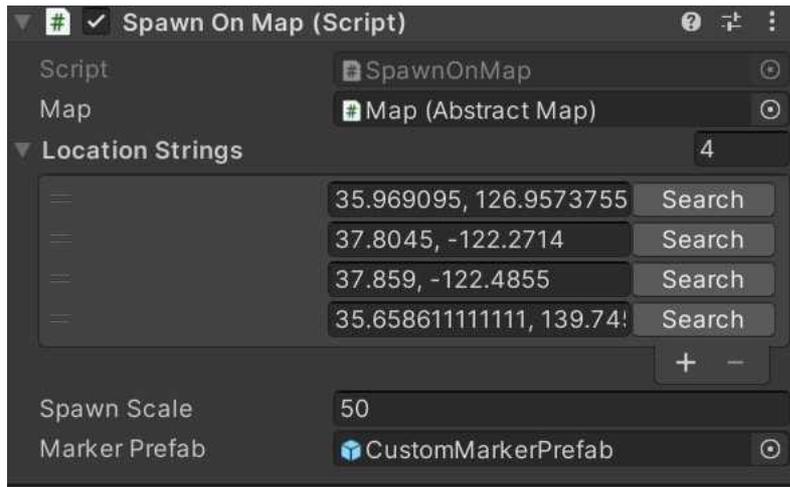
- Abstract Map의 Image/Terrain 옵션
 - 맵의 이미지와 맵의 텍스처를 바꿀 수 있는 옵션



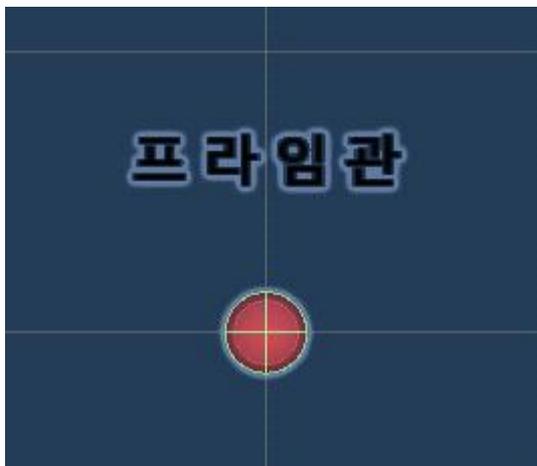
- Quad Tree Camera Movement 스크립트
 - 맵을 터치하고 이동시 이동속도와 확대 속도를 조절 할 수 있음
 - Map Manager에 Map 오브젝트의 Abstract Map의 컴포넌트를 가져와 기능 구현



- Spawn On Map 스크립트
 - 마커를 맵 상에 뿌리는 스크립트
 - 위치 작성 후 프리팹을 가져와 뿌리게 된다.



5. 프리팹 제작



6. 맵 상에 Spawn On Map 스크립트를 적용시킨 결과



7. 최종 결과 창



4) 유저 시나리오 기반의 애니메이션

● 영상에 들어갈 일러스트 제작

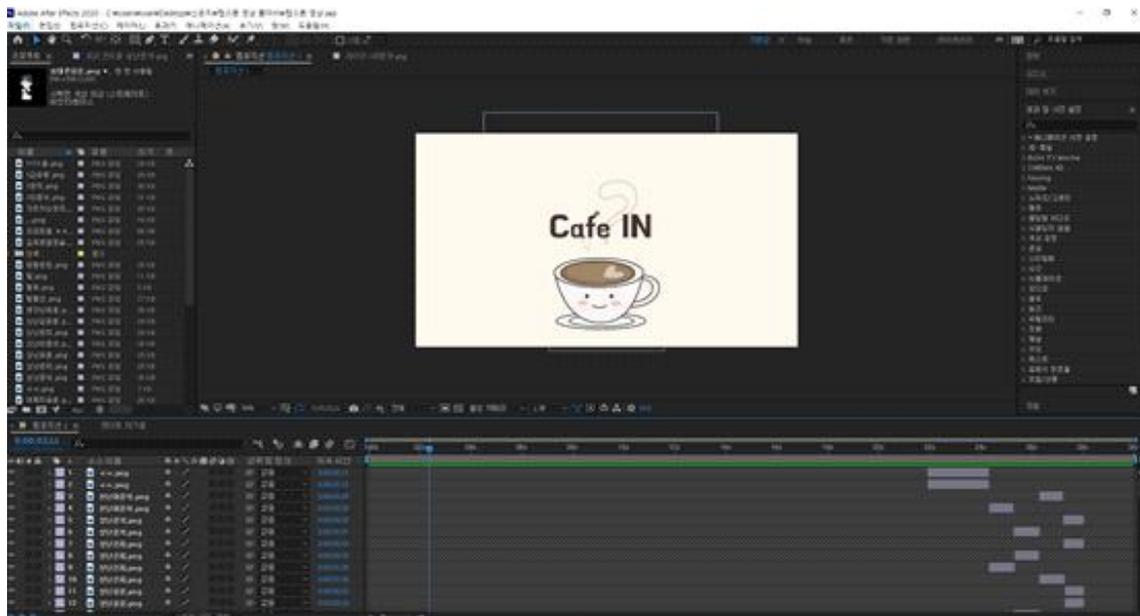
➢ 스토리

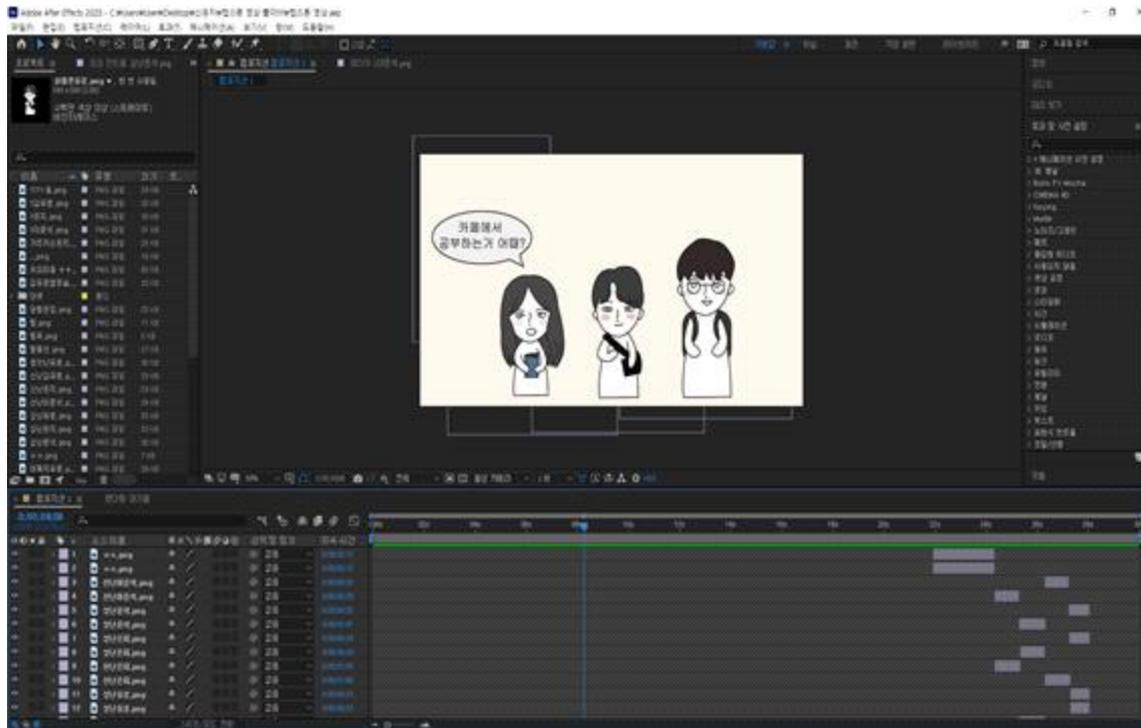
시험기간에 카페에서 공부를 하려고 했는데 카페에 자리가 많아 갈 수 없는 상황 연출
그러다 문득 생각난 앱 'café in'을 사용하는 영상 제작

시험공부하러 카페에 가게되는데 자리가 없어서 카페인 앱을 사용하는 장면



➢ After Effects를 사용한 영상편집





카페에 자리가 없는 스토리





'카페인 앱' 을 활용하는 영상

3. 결과 화면

CafeIN 앱 시작 화면	로그인 화면	회원가입 화면	로그인 성공
			
<p>- CafeIN 캐릭터를 통해 처음 인트로 씬을 보여줌</p>	<p>- 로그인과 회원가입 할 수 있도록 제공함</p>	<p>- 회원가입이 가능하며 Owner 권한은 카페 관리 시스템 등록을 통해 얻을 수 있음</p>	<p>- 아이디와 비밀번호가 맞으면 로그인되어 메인 화면으로 실행됨</p>
지도 맵을 통한 위치 확인	카페 목록 - 별점 순	카페 목록 - 잔여석 순	
			
<p>- 자기 위치와 주변 카페를 확인할 수 있음</p>	<p>- 줌 인/아웃과 위치를 이동할 수 있음</p>	<p>- 별점이 높은 순으로 카페가 정렬됨</p>	<p>- 별점이 잔여석 순으로 카페가 정렬됨</p>

카페 정보 제공		카페로 전화 기능	카페 검색 기능

- 카페의 정보(카페명, 잔여석, 영업시간, 전화번호, 지도, 좌석 배치도)를 얻을 수 있음	- 좌석 문의가 필요할 시, 해당 카페로 전화할 수 있음	- 원하는 카페를 검색하면 카페의 정보가 제공됨
---	---------------------------------	----------------------------

Owner(카페 사장) 기능 - 1. 직접 좌석 업데이트

자신의 카페 관리	카페 잔여석 추가	카페 잔여석 업데이트	카페 잔여석 삭제

- Owner 권한의 아이디를 통해 관리 가능	- 카페의 잔여석을 추가할 수 있음	- 실시간으로 업데이트 되어 카페 리스트로 정렬	- 잔여석 삭제도 가능해 쉽게 관리할 수 있음
---------------------------	---------------------	----------------------------	---------------------------

Owner(카페 사장) 기능 - 2. IoT 좌석벨 연동

좌석벨 블루투스 연결



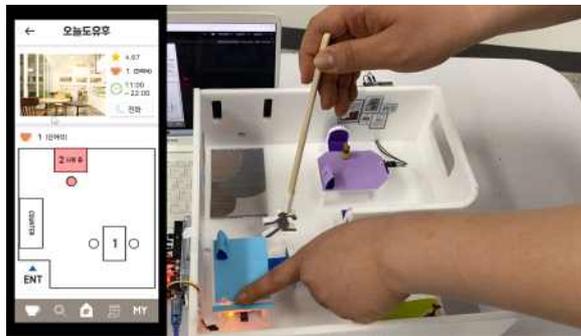
- 카페의 좌석마다 있는 IoT벨은 블루투스로 앱과 연결시킴

1번 좌석벨 ON/OFF



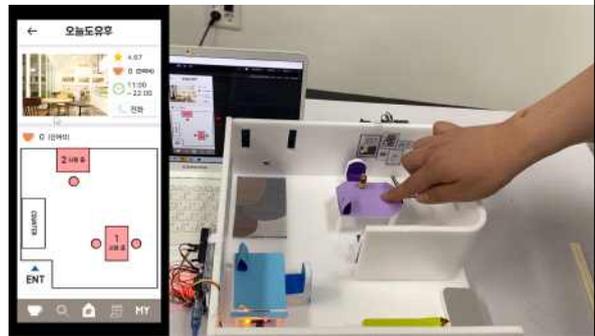
- 1번 자리에 있는 좌석벨이 눌리면 좌석 배치도의 1번 자리가 사용 중으로 켜짐

2번 좌석벨 ON/OFF



- 2번 자리에 있는 좌석벨이 눌리면 좌석 배치도의 2번 자리가 사용 중으로 켜짐

실시간 좌석벨 업데이트



- 카페 이용자가 자리에 앉아 좌석벨을 누르면 실시간으로 ON/OFF가 업데이트 됨