

# 증강현실(AR)을 이용한 학교내부안내 어플리케이션

학생명 : 30412 박한나, 30414 백소현

## I. 서론

### 1. 제작동기 : 학교 내부의 어려움 인식 및 증강현실 제작에 대한 도전

인천공항공고등학교 내부 구조가 어렵다는 것은 학생들 사이에서 많이 언급되어지는 것은 사실이다. 그런데도 불구하고 학교 내부에는 학교 안내도 같은 표지판이 부족하다. 표지판이 설치되어지면 이 문제가 해결될 수 있을까? 이러한 표지판을 설치한다 해도 이를 찾아가기까지가 힘들어 문제해결에는 부족하다는 생각이 들었다. 이 문제에 대해 해결방안을 고려하다 증강현실(AR)을 통한 학교 내부 어플을 만들고자 결심하였다.



[그림1] 증강현실

증강현실이란 사용자가 눈으로 보는 현실세계에 가상 물체를 겹쳐 보여주는 기술이다. 현실세계를 가상세계로 보완해주는 개념인 증강현실은 컴퓨터 그래픽으로 만들어진 가상 환경을 사용하지만 주역은 현실환경이다. 컴퓨터 그래픽은 현실환경에 필요한 정보를 추가 제공하는 역할을 한다. 사용자가 보고 있는 실사 영상에 3차원 가상영상을 겹침(overlap)으로써 현실환경과 가상화면과의 구분이 모호해지도록 한다는 뜻이다.

가상현실기술은 가상환경에 사용자를 몰입하게 하여 실제환경을 볼 수 없다. 하지만 실제환경과 가상의 객체가 혼합된 증강현실기술은 사용자가 실제환경을 볼 수 있게 하여 보다 나은 현실감과 부가 정보를 제공한다. 예를 들어 스마트폰 카메라로 주변을 비추면 인근에 있는 상점의 위치, 전화번호 등의 정보가 입체영상으로 표기된다.

이러한 특성을 가진 증강현실을 이용하여 공항공고등학교 마크를 인식하면 학교내부도를 제공하는 앱을 만들고자 결심하게 되었다.

## 2. 제작목적

### 가. 학생들의 편의증진

인천공항공고등학교 마크를 인식시키면 학교 내부를 입체적으로 제공하는 앱을 개발함을 통해 사용자들이 어디서든 학교 내부에 관한 정보를 제공받을 수 있게 한다. 이로 인한 편의 증진이 기대된다.

### 나. 증강현실에 대한 관심증폭 및 참여 유도

프로그램을 접하는 사용자들은 증강현실을 마주하며 증강현실 제작이 복잡하다는 생각을 전환할 수 있는 계기가 된다. 또한 제작자들 졸업이후 변경되는 학교정보를 후배들이 업데이트 가능하게 해 증강현실에 대한 참여도가 높아짐을 기대할 수 있다.

### 다. 제작자들의 실력향상

복잡한 학교 내부를 컴퓨터 화면에 표현하기 위해서는 여러 프로그램들이 사용된다. 이 과정을 통해 제작자들은 유니티 (Unity 3D), 블렌더(Blender) 등 다양한 프로그램을 접하게 되고 이를 능숙히 사용하기까지 많은 노력을 한다. 이로 인해 제작자들의 실력향상을 기대할 수 있다.

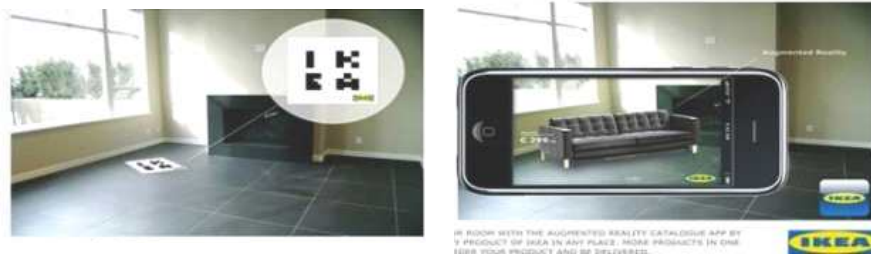
## 3. 이론적 배경

### -증강현실의 기본원리

증강현실은 내부적으로 대단히 복잡하고 어려운 영상 기술이지만, 기본적으로는 다음과 같은 원리와 순서로 작동한다. 증강현실 기술을 적용하기 위해서는 몇 가지 필요한 게 있는데, 지리/위치 정보를 송수신하는 GPS 장치 및 중력(기울기+전자나침반) 센서(또는 자이로스코프 센서), 이 정보에 따른 상세 정보가 저장된 위치정보시스템(인터넷 연결 필요), 그 상세 정보를 수신하여 현실 배경에 표시하는 증강현실 애플리케이션, 마지막으로 이를 디스플레이로 출력할 IT 기기(스마트폰, 태블릿 PC) 등이다.

우선 사용자가 (증강현실 애플리케이션 실행 후) 스마트폰 등의 내장 카메라(캠)로 특정 거리나 건물을 비추면 GPS 수신기를 통해 현재 위치의 위도/경도 정보, 기울기/중력 정보 등이 스마트폰에 임시 기록된다. 그런 다음 이 GPS 정보를 인터넷을 통해 특정 위치 정보 시스템에 전송한다. 해당 위치 반경의 지역이나 건물의 상세 정보를 모두 스마트폰에 저장하기가 현실적으로 불가능하기 때문이다.

사용자로부터 위치/기울기 등의 GPS 정보를 수신한 위치정보시스템은 해당 지역 또는 사물의 상세 정보를 자신의 데이터베이스에서 검색한 후 그 결과를 다시 스마트폰으로 전송한다. 여기에는 물론 특정 건물의 상호, 전화번호 등이 들어 있다. 이 데이터를 수신한 스마트폰은 증강현실 애플리케이션을 통해 현 지도 정보와 매칭시킨 후 실시간 화면으로 보여주는 것이다. 위의 데이터 송수신 단계는 지속적으로 유지, 수행되므로 스마트폰을 들고 거리를 지나면 해당 지역 및 주변에 대한 상세 정보가 순차적으로 화면에 나타나게 된다.



[그림2] 이케아(IKEA)의 증강현실 시스템

-블렌더(Blender)란?

Blender는 원래 네덜란드의 애니메이션 스튜디오인 NeoGeo에서 내부적으로 사용하기 위해 만든 프로그램이었다. 개발사는 Not a Number Technologies이었다. 처음에는 Ton Roosendaal이라는 사람이 주제작자였으나 NaN이라는 회사는 2002년에 부도가 나게 된다. 그래서 Blender의 제작자들은 Blender를 10만 파운드의 기부금을 받게 될 경우 GPL로 코드를 공개하기로 결정하였다. 이 기부금은 빨리 모였고, Blender는 GPL로 공개가 된다. 그 후 Blender는 항상 무료 소프트웨어로 개발되었고, 지금도 2명의 part-time 직원과 2명의 full-time 직원 외에 커뮤니티에 의해서 계속 개발되고 있다.

블렌더(Blender)는 GNU 일반 공중 사용 허가서에 따라 자유 소프트웨어로 릴리즈된 3차원 컴퓨터 그래픽스 소프트웨어이다. 이 프로그램은 모델링, UV 언래핑, 텍스처링, 리깅,

위터 시뮬레이션, 스키닝, 애니메이션, 렌더링, 파티클 등의 시뮬레이션을 수행할 수 있으며 언리니어 편집, 콤포지팅, 파이썬 스크립트 등을 통하여 쌍방향 3차원 프로그램을 제작할 수도 있다.



리눅스, Mac OS X, 마이크로소프트 윈도우 등에서 이용 가능하다.

[그림3] 블렌더 마크(Blender)

-유니티(Unity 3D)란?

유니티(영어: Unity)는 3D 비디오 게임이나 건축 시각화, 실시간 3D 애니메이션 같은 기타 인터랙티브 콘텐츠를 제작하기 위한 통합 저작 도구이다. 에디터는 윈도우와 맥 OS X 상에서 실행되어 윈도우, 맥, Wii, 아이패드, 아이폰 플랫폼으로 게임을 만들 수 있다. 유니티 웹 플레이어 플러그인을 이용하는 웹 브라우저 게임도 제작할 수 있다. 이는 플래시와 유사한 형태이며, 크로스 도메인 보안정책 및 스크립팅에서도 플래시 사용자가 쉽게 적응할 수 있도록 설계되었다.

Unity는 셰이더, 물리 엔진, 네트워크, 지형 조작, 오디오 & 비디오, 애니메이션 등. 게임을 만드는 데 필요한 웬만한 기능을 자체적으로 지원한다. 따라서 초보 개발자도 쉽게 접근할 수 있다. 버튼 하나만 누르면 수많은 플랫폼으로 빌드가 가능하다. 즉, 게임 하나 만들어 놓으면 수많은 플랫폼에서 사용할 수 있단 소리다. 물론, 어느 정도 조정은 필요하다. 유니티가 지원하는 플랫폼은 웹브라우저, 윈도우, 맥, 리눅스, iOS, 안드로이드, 블랙베리, 윈도우 스토어 앱, 윈도우폰, 엑스박스 360, 엑스박스 원, 플레이스테이션 3, 플레이스테이션 4, PS Vita, Wii U, New 닌텐도 3DS, 타이젠, 오클러스 리프트, 구글 카드보드, 스팀 VR, 플레이스테이션 VR, 기어 VR, 홀로렌즈, 데이드림, 안드로이드 TV, 삼성 스마트 TV, tvOS, 닌텐도 스위치, Fire OS, 페이스북 게임룸이 있다. 인디 개발자 혹은 초보자들을 지원하는 프로그램들이 적극적으로 이루어지고 있으며, 이로 인해 신규 진입하는 개발자들이 참고하거나 도움을 받을 만한 곳이 많다는 것도 장점이다.

덕분에 벤처 기업이나 소규모 스타트업 창업, 또는 업종 전환을 하는 사람들 입장에서는 굉장히 유용하게 사용할 수 있다. 기본적으로는 3D로 개발하는 게임 엔진이지만 구현 방식을 조금 손보면 2D 게임도 문제없이 제작할 수 있다.



[그림4] 유니티(Unity) 로고



[그림5]

유니티 설립자들. 왼쪽부터 요하킴 안테(CTO), 니콜라스 프렌시스(CCO), 데이비드 헬가손 (CEO)

## II. 본론

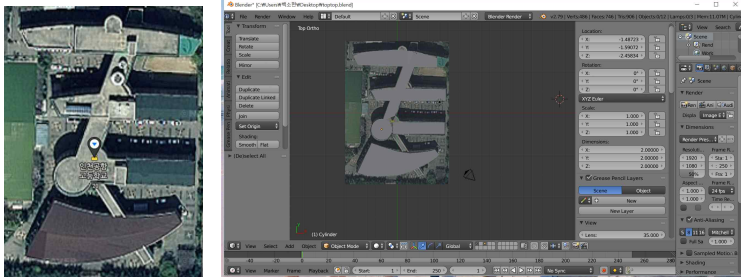
### 1. 제작 과정

-제작기간 :2018.05.03 ~ 제작 중

-사용 프로그램 및 준비물 : 유니티(Unity 3D) + 비주얼 스튜디오(Visual Studio), 블렌더(Blender),

-제작과정

① 인천공항공고등학교 위성사진을 참고, 블렌더(Blender)를 이용해 학교밀면을 모델링한다.



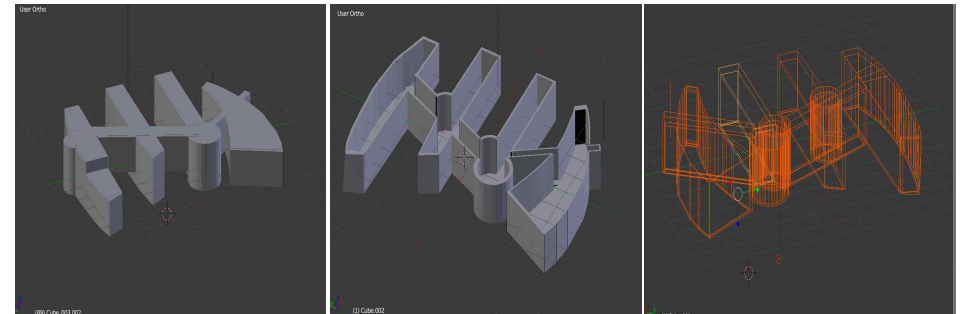
[그림6] 인천공항공고등학교 위성사진/인천고등학교 밀면을 모델링한 모습

② 학교 외부 곳곳을 촬영한다.



[그림7] 인천공항공고등학교 외부 모습

③ 블렌더를 이용해 학교를 모델링한 뒤 단면도로 디자인한다.

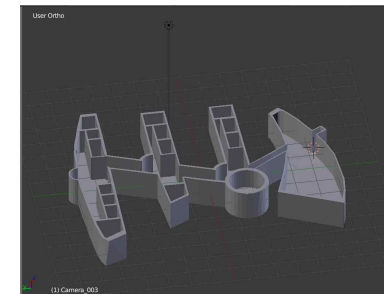


[그림8] 학교 외부 모델링

[그림9] 학교 내부 모델링①

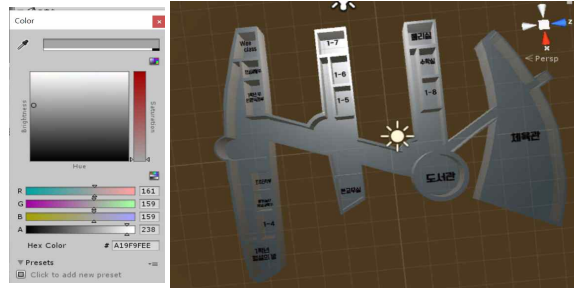
[그림10] 학교 내부 모델링②

④ 학교 내부를 추가적으로 모델링한다.(모든 층 동일적용)



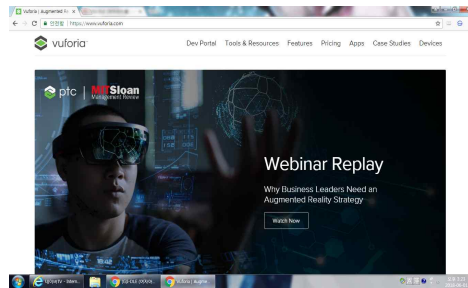
[그림11] 인천공항공고등학교 추가 모델링 모습

⑤ object의 색을 지정한 후 투명도를 높이고 각 내부 구조에 이름을 붙인다.



[그림12] 지정한 색/ light을 쓴 object 모습

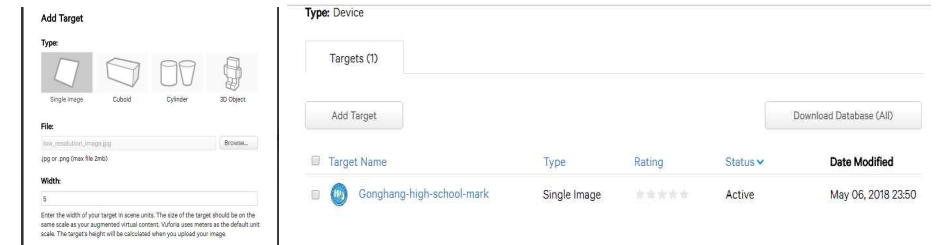
⑥ Vuforia 개발자 포털에서 계정을 만든다. 계정을 만든 후 vuforia의 Unity 패키지를 다운한다. 새 3D 통합 프로젝트를 만든 다음 vuforia Unity 패키지를 열어 패키지의 모든 콘텐츠를 가져온다.



[그림13] Vuforia 포털사이트

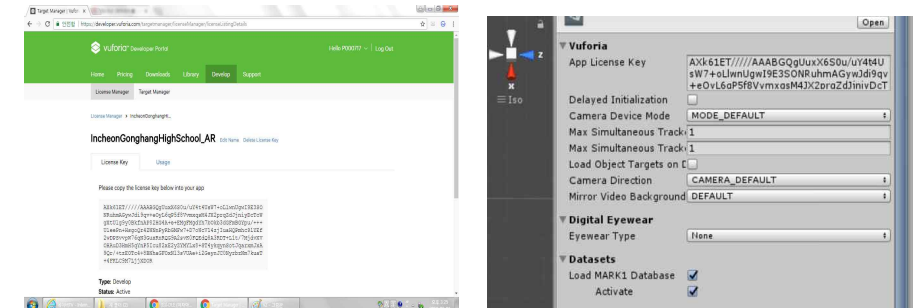
⑦ Vuforia 개발자 포털로 이동해 계정에 로그인 한 후 데이터베이스를 추가한다. 데이터베이스의 이름을 지정하고 유형으로 Device를 선택한다.

⑧ Add Target-Single Image-File에 이미지 선택-Width 지정-Name 지정 한다. 데이터베이스를 다운로드한다.



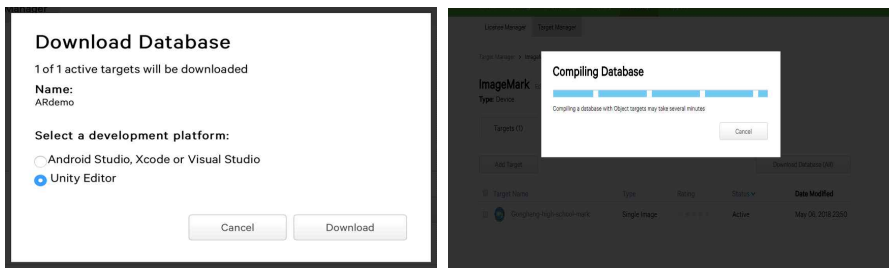
[그림15] Database 를 다운로드 하는 모습

⑨ Assets-Vuforia-Prefabs-AR camera 추가 후 AR Camera-Inspector-App License key Vuforia-계정에 로그인하고 개발 탭에서 라이선스 관리자 섹션을 찾은 후 라이선스 키를 추가 한다. 이 라이선스를 복사 한 후 AR Camera 설정의 App License Key 섹션에 붙여 넣는다.

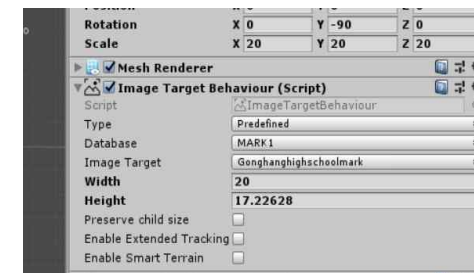


[그림16] 라이선스를 추가하는 모습

⑩ Assets-vuforia-Prefabs Image Target 개체를 장면에 추가한다.



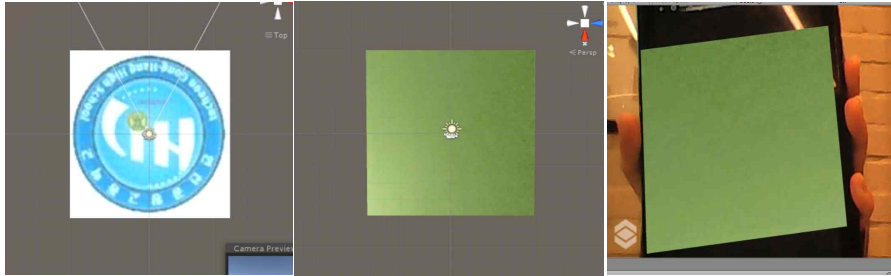
[그림14] Database를 추가하는 모습



[그림17] Image Target 추가하는 모습



⑪ Image Target 위에 3D-plane을 import 한 후 인천공항고등학교 잔디를 텍스처 (Texture)로 적용시킨다.

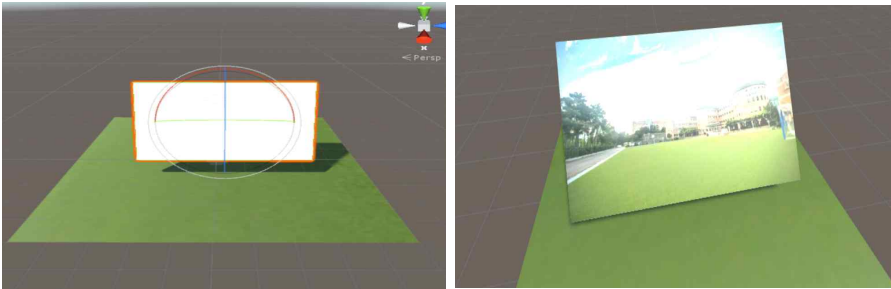


[그림18] 학교 마크를 인식시킨 모습

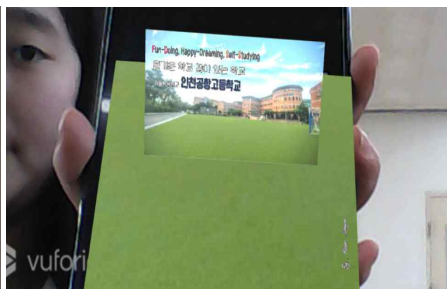
[그림19] Plane을 덮은 후 텍스처를 적용 한 모습

[그림20] AR카메라에서 인식되어지는 모습

⑫ Plane위에 3D Object-Cube를 Import 한다. 공항고등학교 외부 전경을 Texture로 지정하고 그림글씨를 Cube위에 적절히 배치한 후 AR Camera에 인식시킨다.

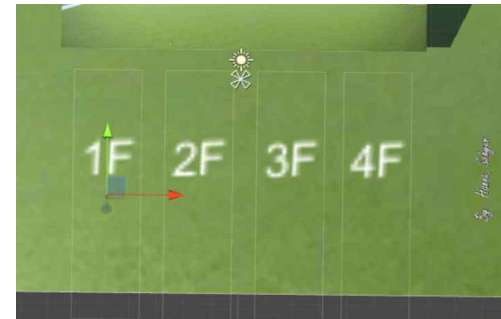


[그림22] 글씨를 배치한 모습

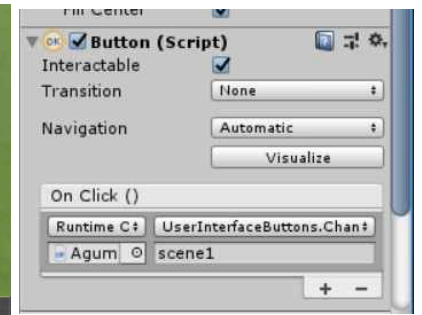


[그림23] Ar Camera에 인식시킨 모습

⑬ 각 층 버튼을 배치한다.



[그림24] 버튼을 배치한 모습



[그림25] 버튼에 scene을 추가한 모습

⑭ Assets 에 폴더를 만든다. 스크립트 파일을 이 폴더에 저장합니다. 그런 다음 이 폴더에 C# 스크립트 파일을 만듭니다. 이름을 "AgumentedReality\_UI"로 지정한다.

```
using UnityEngine;
using System.Collections;
using System.IO;

public class UserInterfaceButtons : MonoBehaviour
{
    public float scalingSpeed = 0.03f;
    public float rotationSpeed = 70.0f;
    public float translationSpeed = 5.0f;
    public GameObject Model;
    //
    bool repeatScaleUp = false;
    bool repeatScaleDown = false;
    bool repeatRotateLeft = false;
    bool repeatRotateRight = false;
    bool repeatPositionUp = false;
    bool repeatPositionDown = false;
    bool repeatPositionLeft = false;
    bool repeatPositionRight = false;

    void Update ()
    {
        if (repeatScaleUp) {
            ScaleUpButton ();
        }
        if (repeatScaleDown) {
            ScaleDownButton ();
        }
        if (repeatRotateRight) {
            RotationRightButton();
        }
        if (repeatRotateLeft) {
            RotationLeftButton();
        }
        if (repeatPositionUp) {
            PositionUpButton();
        }
        if (repeatPositionDown) {
            PositionDownButton();
        }
    }

    public void PositionDownButtonRepeat ()
    {
        repeatPositionDown = true;
    }
    public void PositionUpButtonRepeat ()
    {
        repeatPositionUp = true;
    }
    public void PositionLeftButtonRepeat ()
    {
        repeatPositionLeft = true;
    }
    public void PositionRightButtonRepeat ()
    {
        repeatPositionRight = true;
    }

    public void ScaleUpButtonOff ()
    {
        repeatScaleUp = false;
        Debug.Log ("Off");
    }
    public void ScaleDownButtonOff ()
    {
        repeatScaleDown = false;
        Debug.Log ("Off");
    }

    public void RotateLeftButtonOff ()
    {
        repeatRotateLeft = false;
        Debug.Log ("Off");
    }

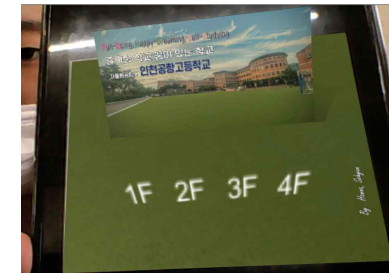
    public void RotateRightButtonOff ()
    {
        repeatRotateRight = false;
        Debug.Log ("Off");
    }

    public void PositionRightButtonOff ()
    {
        repeatPositionRight = false;
        Debug.Log ("Off");
    }
}
```

<pre>         if (repeatPositionLeft) {             PositionLeftButton();         }          if (repeatPositionRight) {             PositionRightButton();         }      }      public void CloseAppButton ()     {         Application.Quit ();     }      public void RotationRightButton ()     {         // transform.Rotate (0,         -rotationSpeed * Time.deltaTime, 0);         G a m e O b j e c t . F i n d W i t h T a g         ("Model").transform.Rotate (0, -rotationSpeed *         Time.deltaTime, 0);     }      public void RotationLeftButton ()     {         // transform.Rotate (0,         rotationSpeed * Time.deltaTime, 0);         G a m e O b j e c t . F i n d W i t h T a g         ("Model").transform.Rotate (0, rotationSpeed *         Time.deltaTime, 0);     }      public void RotationRightButtonRepeat ()     {         // transform.Rotate (0,         -rotationSpeed * Time.deltaTime, 0);         repeatRotateRight=true;     }      public void RotationLeftButtonRepeat ()     {         // transform.Rotate (0,         rotationSpeed * Time.deltaTime, 0);         repeatRotateLeft=true;     }      public void ScaleUpButton ()     {         // transform.localScale += new         Vector3(scalingSpeed, scalingSpeed, scalingSpeed);         G a m e O b j e c t . F i n d W i t h T a g         ("Model").transform.localScale += new Vector3         (scalingSpeed, scalingSpeed, scalingSpeed);     }      public void ScaleUpButtonRepeat ()     {         repeatScaleUp = true;         Debug.Log ("Up");     }      public void ScaleDownButtonRepeat ()     {         repeatScaleDown = true;         Debug.Log ("Down");     } </pre>	<pre>     }      public void PositionLeftButtonOff ()     {         repeatPositionLeft = false;         Debug.Log ("Off");     }      public void PositionUpButtonOff ()     {         repeatPositionUp = false;         Debug.Log ("Off");     }      public void PositionDownButtonOff ()     {         repeatPositionDown = false;         Debug.Log ("Off");     }      public void ScaleDownButton ()     {         // transform.localScale += new         Vector3(-scalingSpeed, -scalingSpeed,         -scalingSpeed);         G a m e O b j e c t . F i n d W i t h T a g         ("Model").transform.localScale += new Vector3         (-scalingSpeed, -scalingSpeed, -scalingSpeed);     }      public void PositionUpButton ()     {         G a m e O b j e c t . F i n d W i t h T a g         ("Model").transform.Translate (0, 0, -translationSpeed         * Time.deltaTime);     }      public void PositionDownButton ()     {         G a m e O b j e c t . F i n d W i t h T a g         ("Model").transform.Translate (0, 0, translationSpeed         * Time.deltaTime);     }      public void PositionRightButton ()     {         G a m e O b j e c t . F i n d W i t h T a g         ("Model").transform.Translate (-translationSpeed *         Time.deltaTime, 0, 0);     }      public void PositionLeftButton ()     {         G a m e O b j e c t . F i n d W i t h T a g         ("Model").transform.Translate (translationSpeed *         Time.deltaTime, 0, 0); // backward     }      public void ChangeScene(string a)     {         #pragma warning disable CS0618 // Type or member         is obsolete         Application.LoadLevel(a);         #pragma warning restore CS0618 // Type or member         is obsolete     }      public void AnyButton ()     {         Debug.Log ("Any");     } </pre>
--	--

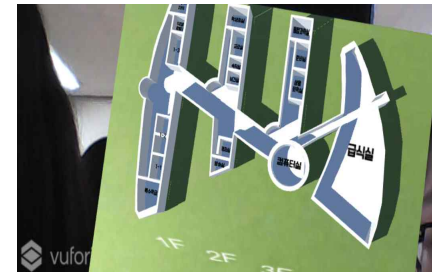
-탐구결과

1. 카메라에 인천공항고등학교 마크를 인식시킨다. 1F 버튼을 눌러본다.

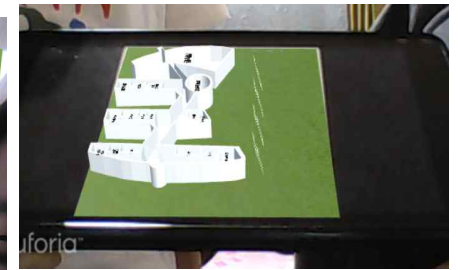


[그림26] 공항고 마크를 인식 시킨 모습

다음과 같은 화면이 나온다.



[그림27] 정면에서 본 1층



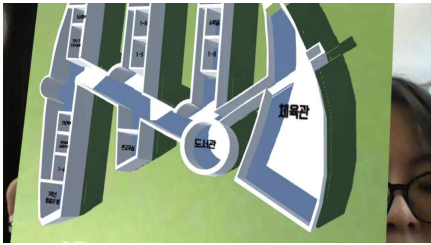
[그림28] 측면에서 본 1층

2. 카메라에 인천공항고등학교 마크를 인식시킨다. 2F 버튼을 눌러본다.



[그림29] 공항고 마크를 인식시킨 모습

다음과 같은 화면이 나온다.

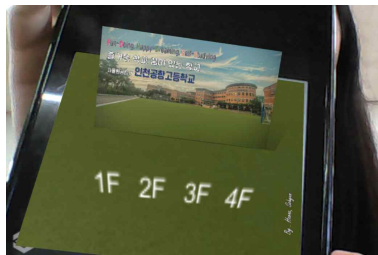


[그림30] 정면에서 본 2층



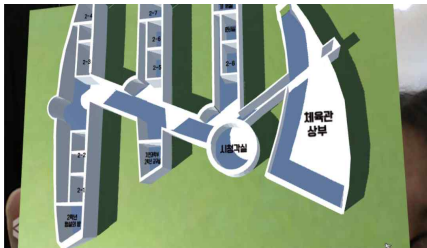
[그림31] 측면에서 본 2층

3. 카메라에 인천공항고등학교 마크를 인식시킨다. 3F 버튼을 눌러본다.



[그림32] 공항고 마크를 인식시킨 모습

다음과 같은 화면이 나온다.

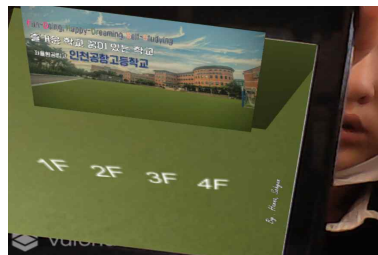


[그림33] 정면에서 본 3층



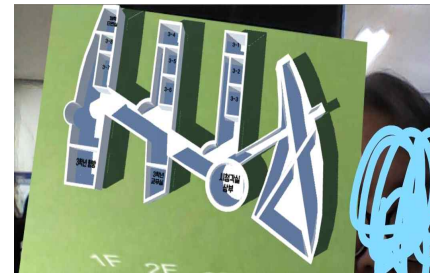
[그림34] 측면에서 본 3층

4. 카메라에 인천공항고등학교 마크를 인식시킨다. 4F 버튼을 눌러본다.

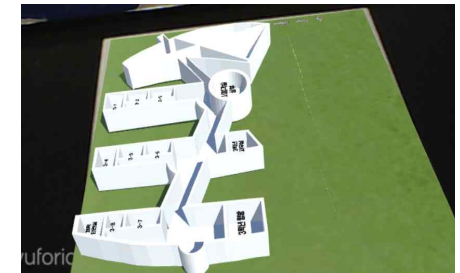


[그림35] 공항고 마크를 인식시킨 모습

다음과 같은 화면이 나온다.



[그림36] 정면에서 본 4층



[그림37] 측면에서 본 4층

## 2. 선행 작품 및 참고자료

- Unity+vuforia+animated 3D AR 앱 만들기  
[https://youtu.be/0SwdshyV\\_cg](https://youtu.be/0SwdshyV_cg)

- Unity5 첫 AR Project 만들기  
<https://youtu.be/MyjTQ9EKil4>

- How To Augmented Reality App Tutorial Virtual Buttons with Unity and Vuforia  
<https://youtu.be/Fgd21lbhikU>

- 블렌더Blender)강좌  
<https://www.youtube.com/channel/UCCVtEH6dckRBwr751Hz7Iyw>

- 인천공항고등학교 홈페이지  
<http://gonghang-h.icehs.kr/>

- 인천공항고등학교 16대 학생회 페이스북  
[https://www.facebook.com/인천공항고등학교-학생회-16대-1092369114183396/?hc\\_ref=ARSCofdvGh6oivDiyu41MSqt9cDhv\\_oE9opNEgwX3UUYbH0kuQikmciz0vVVeStYQzk](https://www.facebook.com/인천공항고등학교-학생회-16대-1092369114183396/?hc_ref=ARSCofdvGh6oivDiyu41MSqt9cDhv_oE9opNEgwX3UUYbH0kuQikmciz0vVVeStYQzk)

-AR Projects

<https://www.youtube.com/watch?v=kYX-DMUbeeE>

-증강현실 사전

<https://terms.naver.com/entry.nhn?docId=3573450&cid=59088&categoryId=59096>

<https://terms.naver.com/entry.nhn?docId=1227952&cid=40942&categoryId=32828>

<https://terms.naver.com/entry.nhn?docId=3408914&cid=58413&categoryId=58413>

### Ⅲ. 결론 및 제언

올해까지 횡수로 3년째 다니고 있는 인천공항공고등학교, 학교에 IT 관련 계열 꿈을 품고 입학 했을 때 까지만 해도 꿈은 단지 꿈일 뿐이었다. 간절히 원하는 장래희망이긴 하지만 그에 관한 지식은 ‘무’에 가까웠기 때문이다. 이후 어떻게 보면 약간은 멍청하게, 그렇게 처음부터 하나하나 배우기 시작했다. 그런 우리들이 이번 프로젝트를 이만큼 진행 시키기까지는 정말 많은 노력이 있었다. 작품을 만들기 위해 필요한 프로그램을 배우고, 관련 지식을 섭렵하는 과정까지의 반복. 사실 이 프로젝트는 작년에 시도하다 실패한 프로젝트였다. 나름대로 열심히 배웠지만 프로젝트를 진행하기에는 수준 미만인 실력에 좌절하며 프로젝트를 포기했던 그때, 그 아쉬움이 너무 커서 이번년도에 다시 도전하고자 결심하게 되었다. 다시 시작한 프로젝트는 놀랍도록 수월히 진행 되었다. 방학기간 동안 익숙해질 때 까지 손에서 놓지 않고 다루었던 블렌더(blender)와 유니티 3D (Unity 3D), 그 노력이 빛을 발한 듯 했다. 작년의 실패가 이제는 아쉽지 않고 오히려 학교를 졸업할 무렵인 고등학교 3학년 시기에 본교에 대해서 자세히 탐구해야만 하는 프로젝트를 진행할 수 있게 되어 의미있다고 생각한다. 또한 실패에 대해 포기하지 않고 극복해 나가는 과정을 거치면서 한층 더 성장할 수 있는 기회를 가지게 된 것 같다.

증강현실(AR)을 통한 학교내부제공프로그램 제작이 처음의 목표였지만 아쉽게도 아직 완벽히 성공하지는 못한 것 같다. 유니티(Unity 3D) 프로그램 내에서 플레이가 가능하도록 구현했지만 이를 어플리케이션의 형태로 빌드하는 과정에서 오류가 반복되어 어려움을 겪고 있다. 이후 오류를 해결하기 위해 지금까지도 여러 외국 사이트를 방문하고, 질문을 올리는 등의 끊임없는 노력을 하고 있지만 아직 미해결상태이다. 사실 학술제 준비 기간은 하나의 프로그램을 만들기엔 턱 없이 부족하고 짧은 기간이라 처음부터 기간내에 완성할 수 있을 것이라 생각하진 않았다. 그렇기에 프로젝트 진행률에 대해서는 만족하고있다. 또한 학술제 이후 시간의 여유를 가지고 프로젝트를 계속 진행해 나가면 오류를 해결할 수 있다는 강한 확신을 가지고 있기에 앞으로의 계획에 대해서도 긍정적인 전망이라 예상한다.