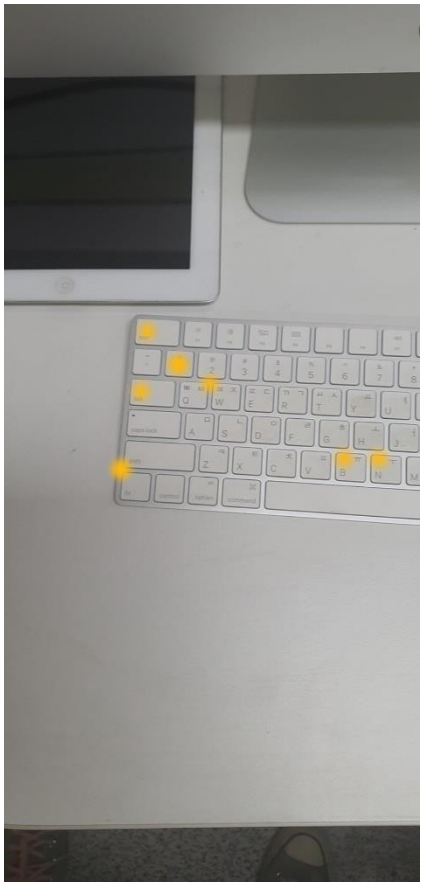
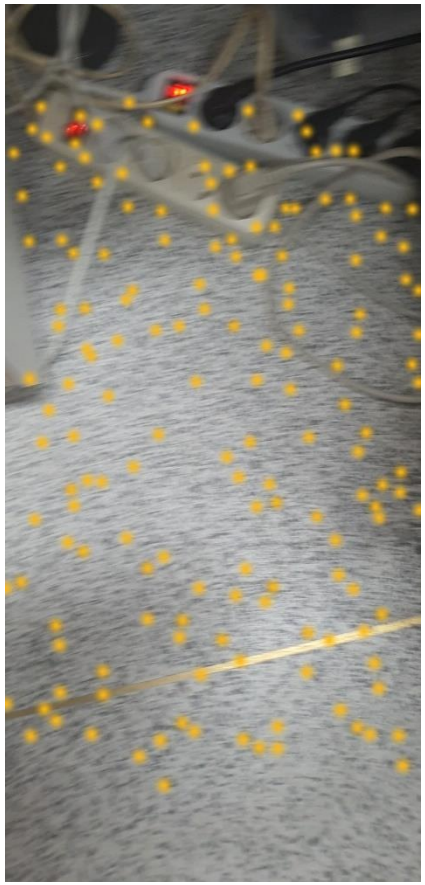
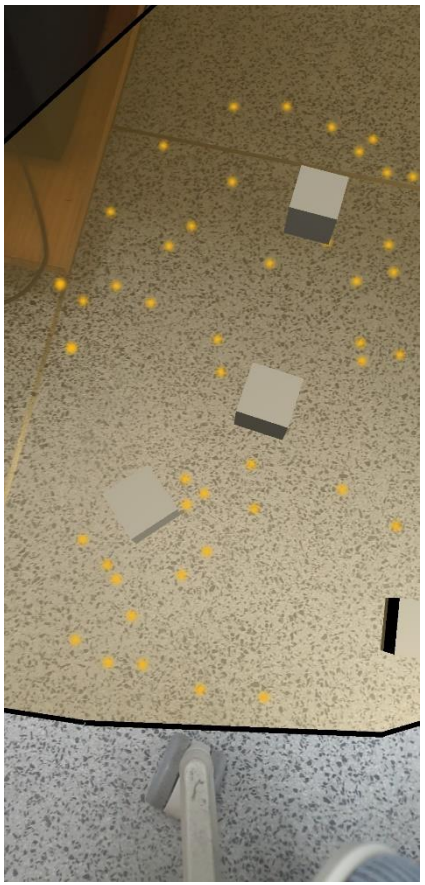
ARFoundation

Part 1.

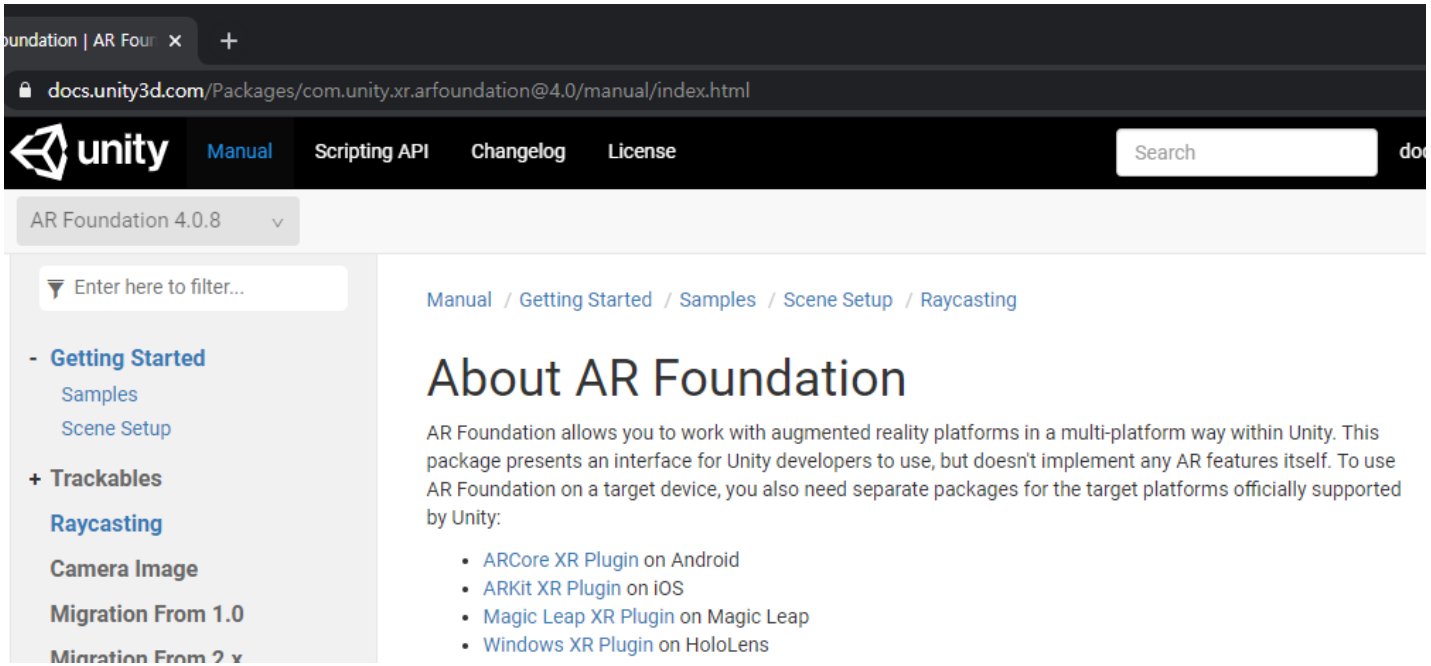
Final Result Preview

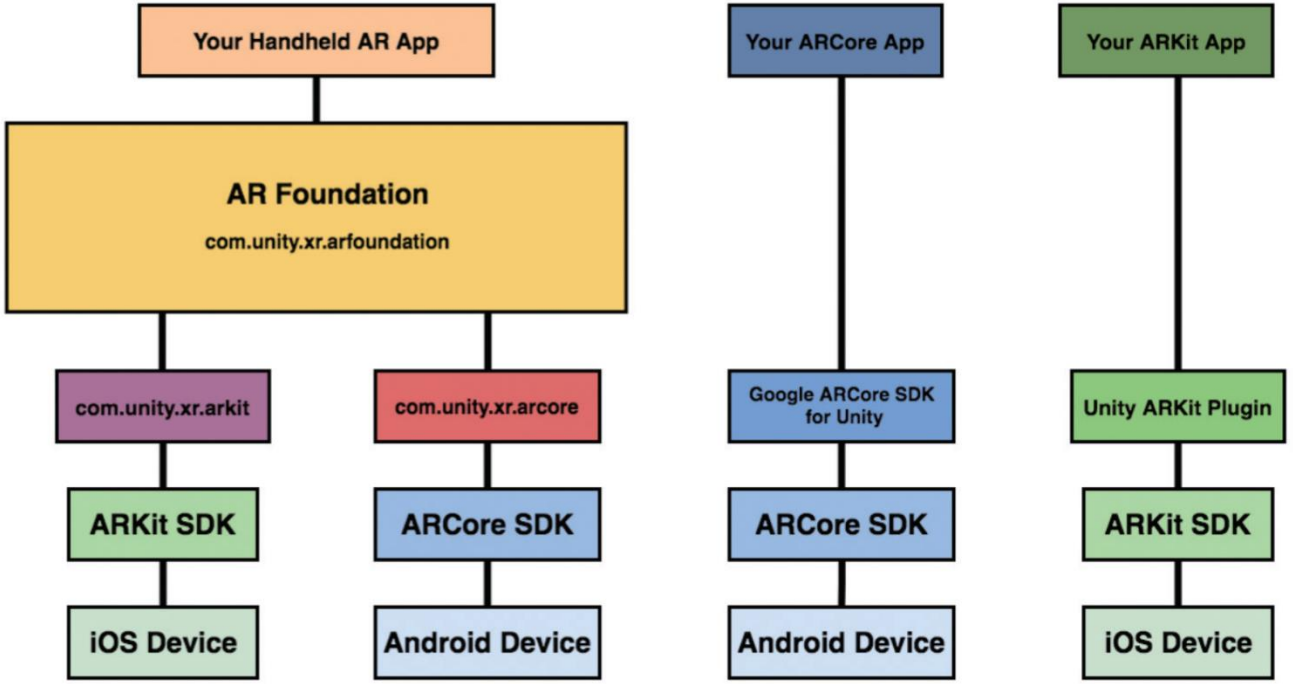


학습내용

1. ARFoundation Import
2. Run Sample App



- ARFoundation
  - 멀티플랫폼 핸드헬드 AR 패키지
  - ARCore와 ARKit의 코어 SDK를 바탕으로, Android 또는 iOS로 빌드 기능을 제공
  - 단일화된 개발 SDK로, 멀티플랫폼으로 빌드할 수 있는 장점



– ARFoundation 2.1.0 버전의 주요 지원 기능

지원되는 기능	AR Foundation	ARCore SDK	ARKit SDK
수평 평면 검출	✓	✓	✓
수직 평면 검출	✓	✓	✓
특징점(Feature Point) 검출	✓	✓	✓
레이캐스팅	✓	✓	✓
이미지 추적	✓ (2.1.0부터 지원)	✓	✓
3D 오브젝트 추적	✓ (2.1.0부터 지원)	–	✓
환경 프로브	✓ (2.1.0부터 지원)	–	✓
월드 맵	✓	–	✓
얼굴 추적	✓ (2.1.0부터 지원)	–	✓
클라우드 앵커	개발 중	✓	–
에디터 리모팅	개발 중	✓ (인스턴트 프리뷰)	✓ (ARKit 리모트)
에디터 시뮬레이터	개발 중	–	–
LWRP 지원	3.3.0 지원	개발 중	개발 중
카메라 이미지 접근 API	✓	–	–

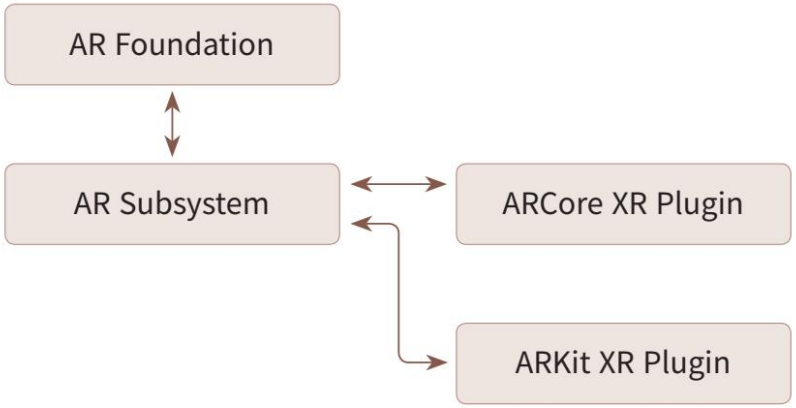
– ARFoundation 4.0.8 – Feature support per platform

	ARCore	ARKit	Magic Leap	HoloLens
Device tracking	✓	✓	✓	✓
Plane tracking	✓	✓	✓	
Point clouds	✓	✓		
Anchors	✓	✓	✓	✓
Light estimation	✓	✓		
Environment probes	✓	✓		
Face tracking	✓	✓		
2D Image tracking	✓	✓	✓	
3D Object tracking		✓		
Meshing		✓	✓	✓
2D & 3D body tracking		✓		
Collaborative participants		✓		
Human segmentation and occlusion		✓		
Raycast	✓	✓	✓	
Pass-through video	✓	✓		
Session management	✓	✓	✓	✓

About ARFoundation

– 개발 환경

- 코어 SDK(ARCore, ARKit)을 기반으로 구성된 시스템 : 서브시스템
- 코어 SDK의 저수준 API를 사용해 통신하는 것은 AR 서브시스템이 처리
- ARFoundation은 고수준 API로 개발환경을 제공 : ARCore XR Plugin 또는 ARKit XR Plugin 중 하나는 반드시 설치해야 함



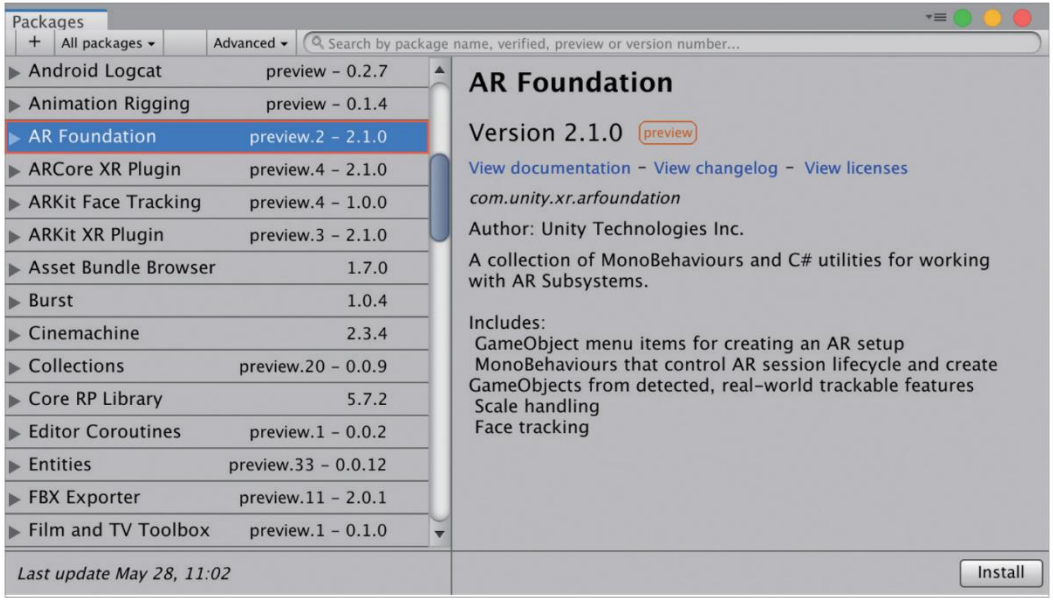
[그림 11-2] AR Foundation의 구조

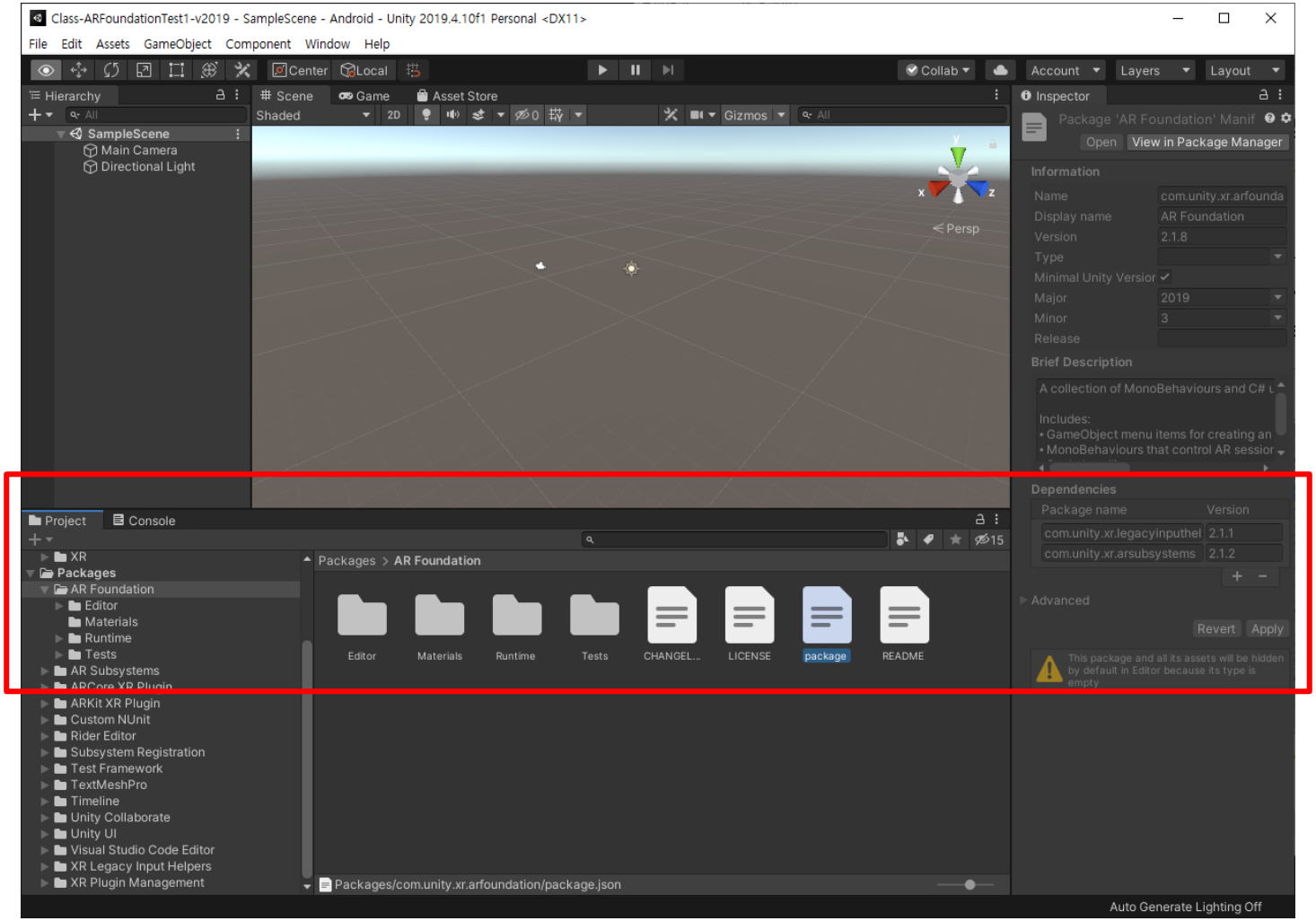
– 프로젝트 환경 설정

Getting started
with ARFoundation

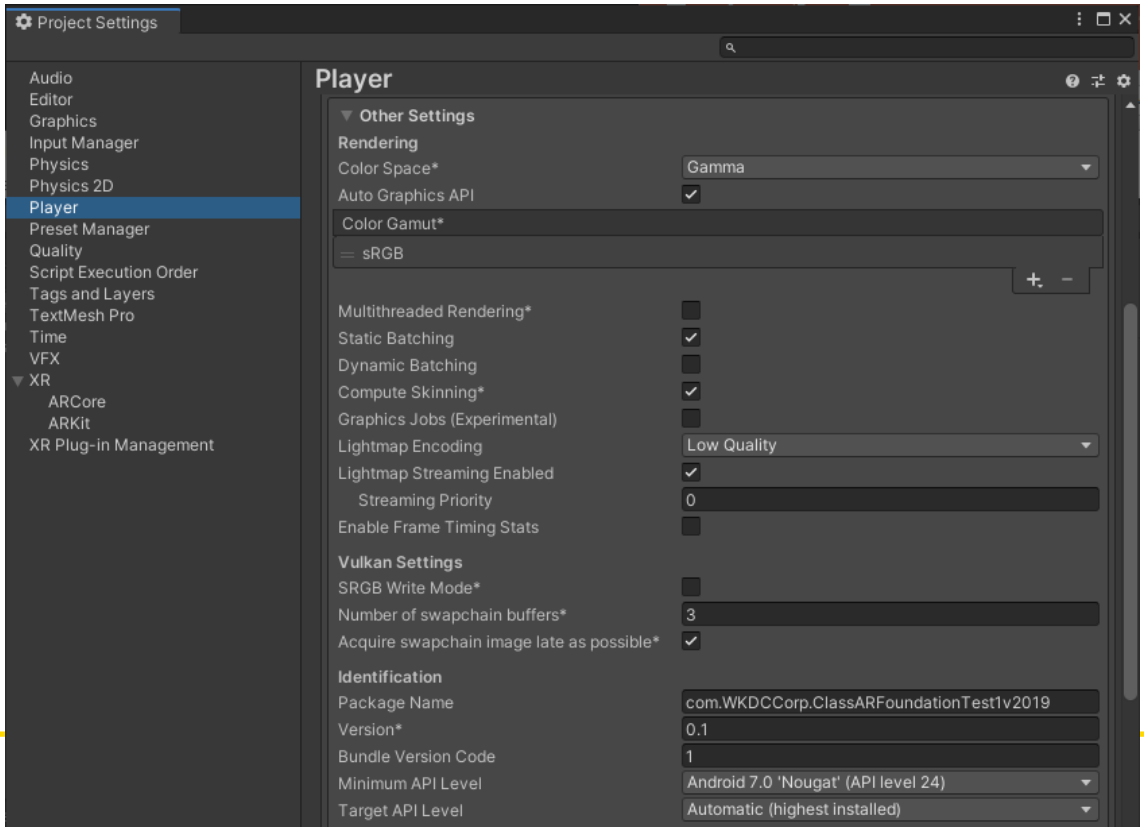
- 새로운 프로젝트를 생성하고 플랫폼을 Android로 변경
- [Windows – Package Manager]을 클릭하여 패키지 매니저를 오픈
- [Advanced – Show preview packages]를 선택하여 패키지에서 ARFoundation과 ARCore XR Plugin을 설치

의존성 있는 2개의 패키지가 함께 설치됨 (AR Subsystems, XR Legacy Input Helpers)

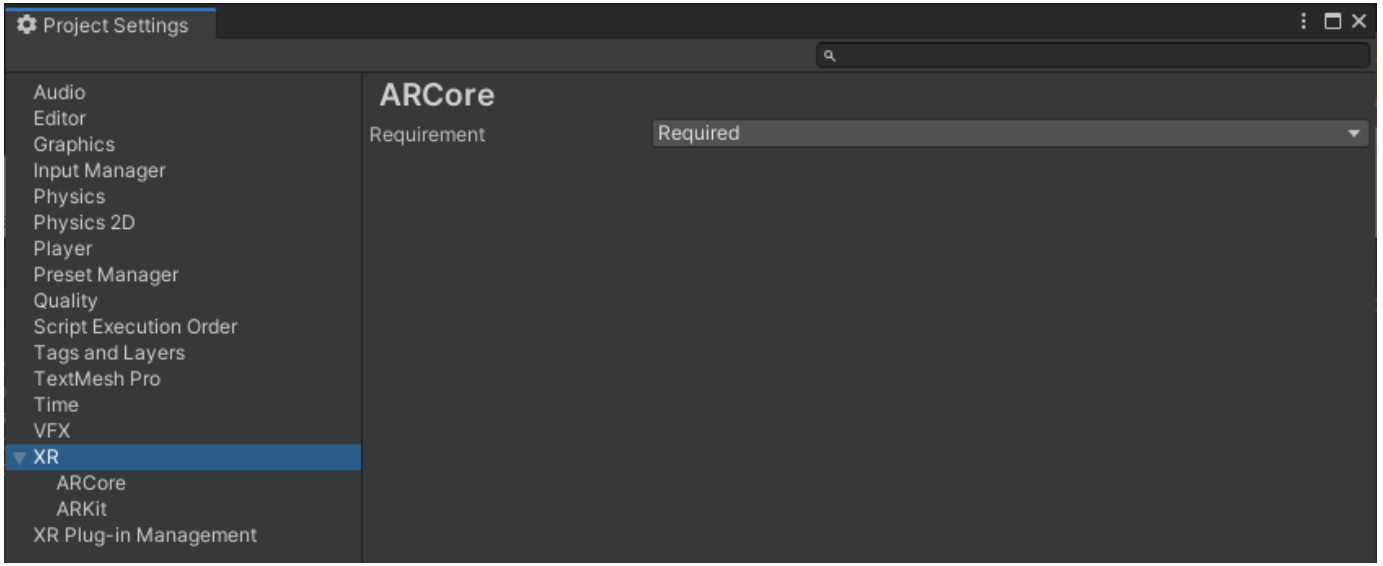




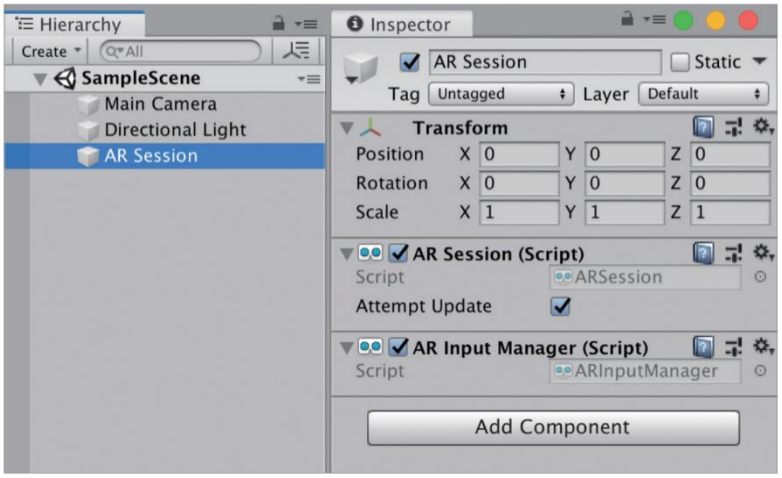
- 프로젝트 환경 설정 (for Android)
  - [Edit – Project Settings]를 클릭하고 Player 탭을 선택
  - Product Name을 입력 (디바이스에 나오는 이름임)
  - Other Setting 탭을 클릭하여 아래 항목을 설정
    - » Auto Graphics API 체크
    - » Multithread Rendering 체크 해제
    - » Package Name을 유일한 값으로 설정 (예, com.companyName.productName)
    - » Minimum API Level을 Android 7.0 Nougat (API level 24)로 설정

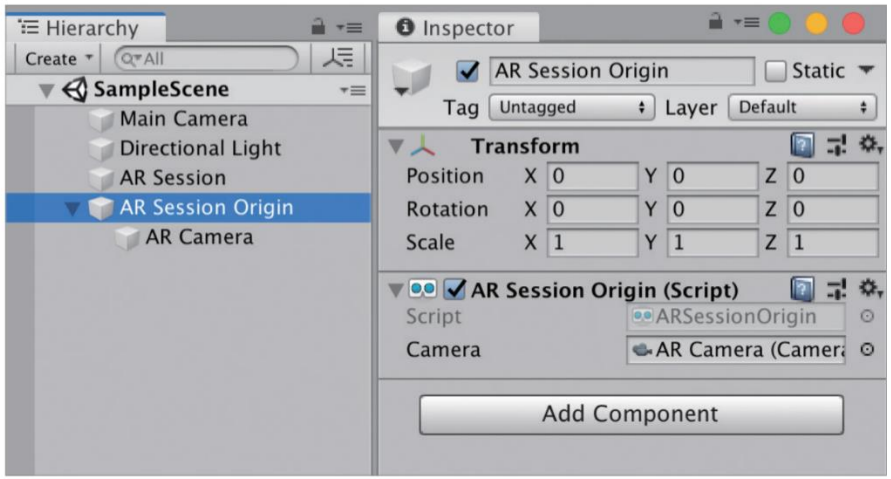


– 좌측 메뉴에서 XR을 선택하고 [Create] 버튼을 클릭하여 ARCore를 위한 애셋을 설정

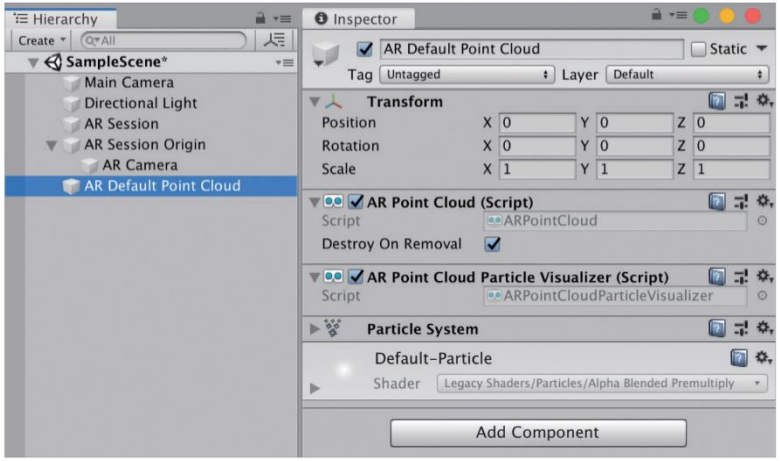


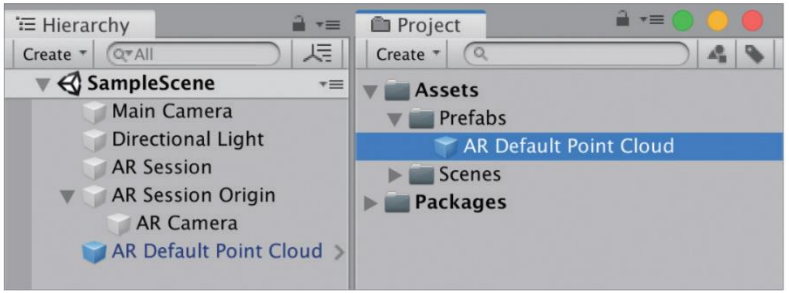
- 포인트 클라우드 및 평면 인식
  - [GameObject – XR – AR Session]을 클릭
    - AR Session은 AR 환경의 설정 및 라이프사이클을 조절하는 역할 (씬에 하나만 존재해야 함)
  - [GameObject – XR – AR Session Origin]을 클릭
    - AR Session Origin은 하위에 현실세계를 비추는 AR Camera를 포함하고 있으면 실행 시 디바이스의 위치를 월드좌표의 중심(0,0,0)으로 인식
    - 감지한 Feature Point에 생성한 모든 게임오브젝트(평면 매쉬, 포인트 클라우드)는 AR Session Origin의 하위로 차일드화된다



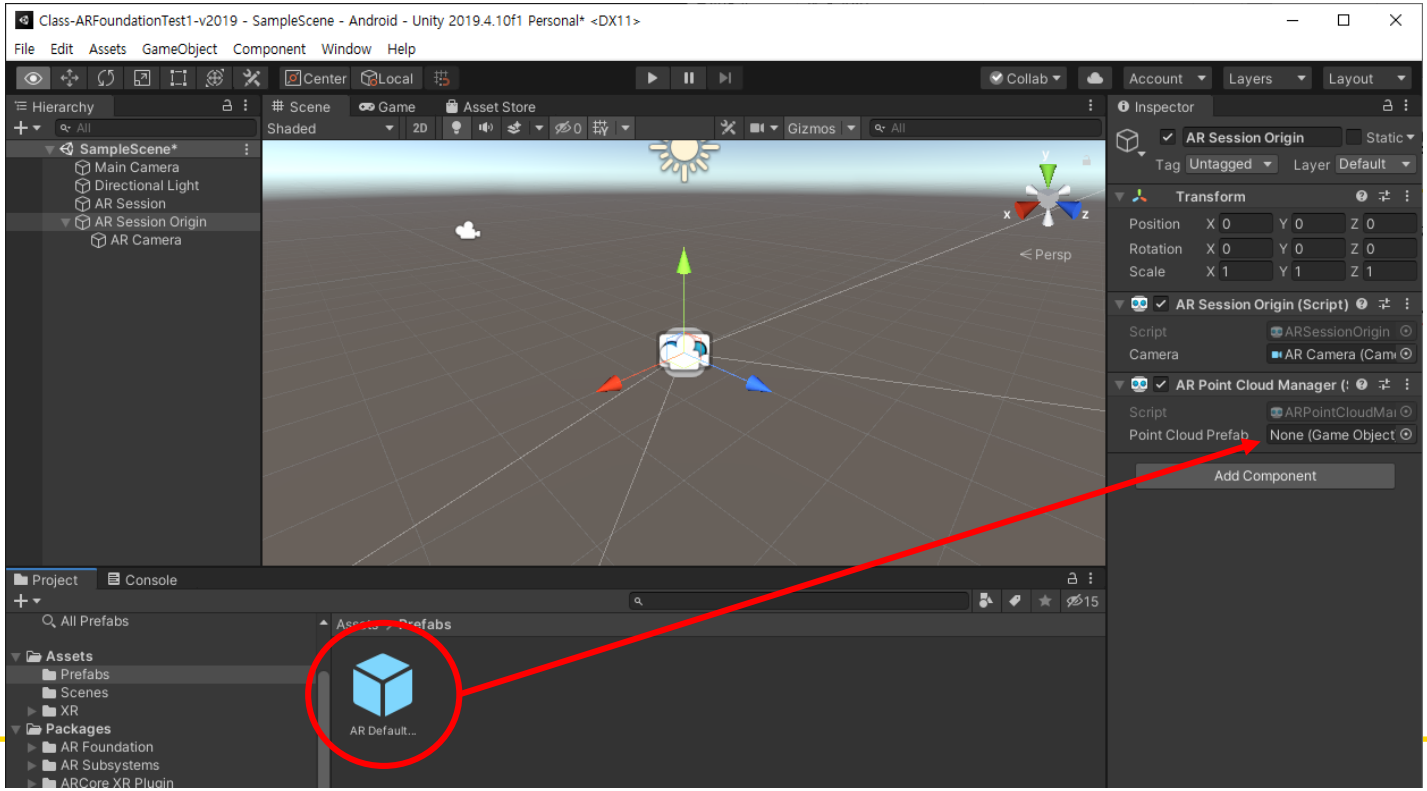


- [GameObject – XR – AR Default Point Cloud]를 클릭
  - AR Default Point Cloud는 AR Camera의 영상으로 감지한 Feature Point를 표시
  - 생성된 AR Default Point Cloud를 프로젝트 뷰로 드래그하여 프리팹으로 만듦
  - 프리팹으로 전환한 다음, 하이러리키 뷰에서는 AR Default Point Cloud를 삭제



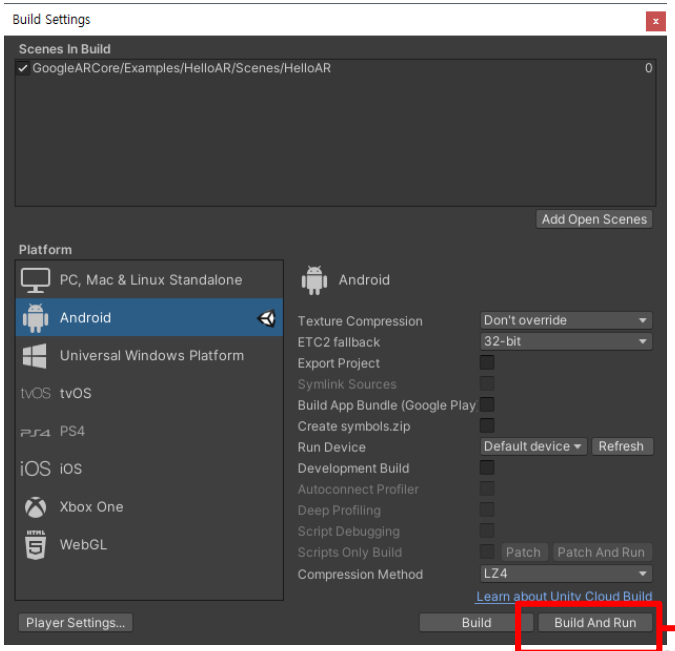


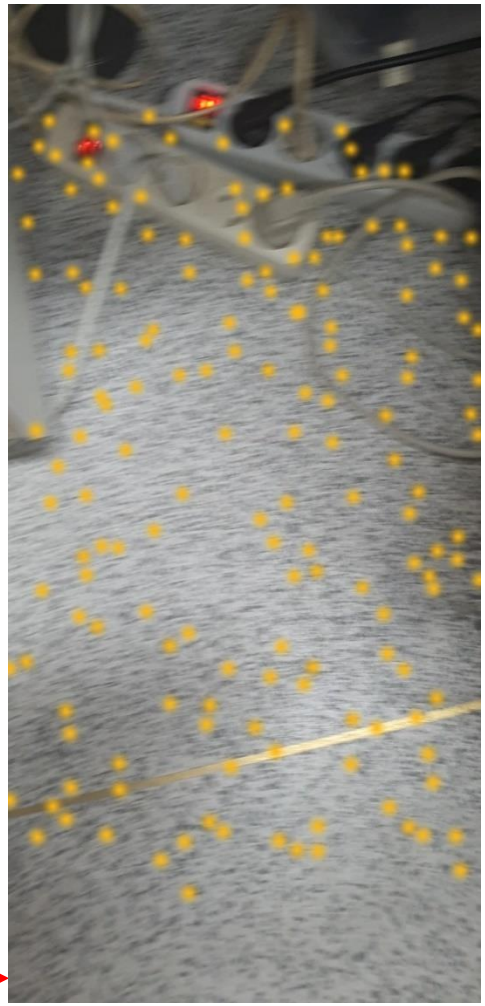
- Feature Point를 감지하기 위해 AR Session Origin에 AR Point Cloud Manager 스크립트를 추가
  - 하이러리키 뷰에서 AR Session Origin을 선택하고 인스팩터 뷰에서 [Add Component] 버튼을 클릭하여 해당 스크립트(AR Point Cloud Manager)를 추가
  - 생성한 AR Default Point Cloud 프리팹을 드래그하여 연결



Build and run the app

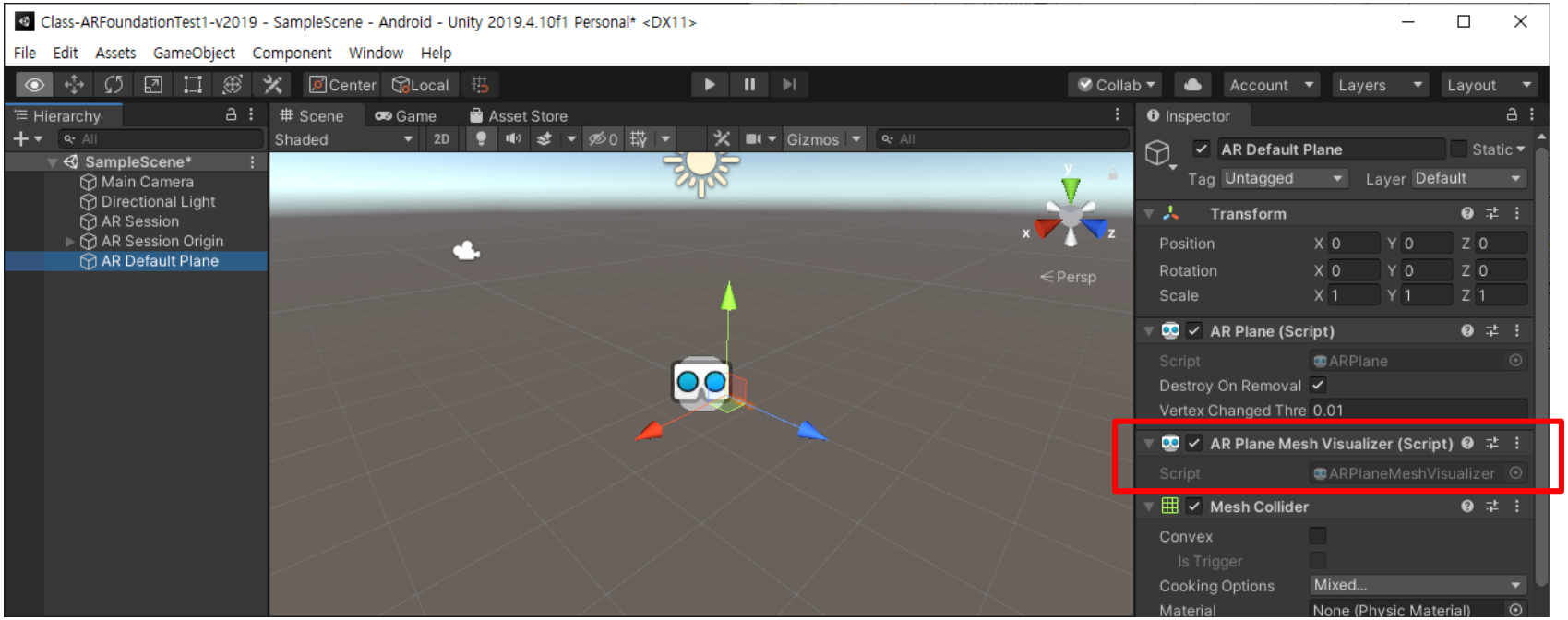
1. 스마트폰에서 [개발자 옵션 및 USB 디버깅]을 활성화
2. 디바이스를 컴퓨터에 연결
3. Building Settings 창에서 [Build and Run] 버튼 클릭



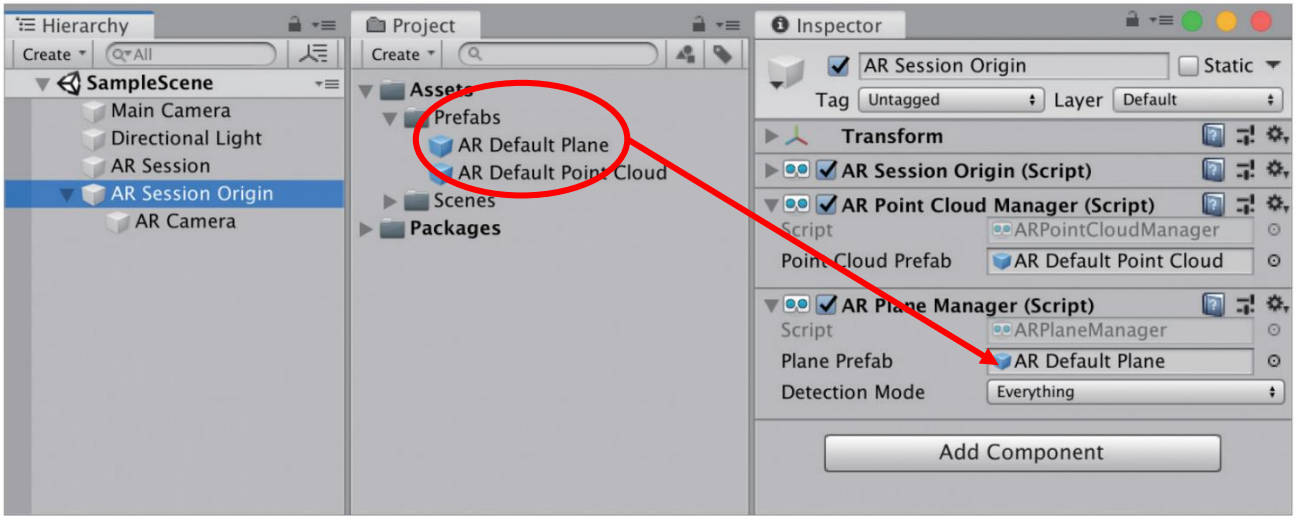


AR_Foundation.zip

- [GameObject – XR – AR Default Plane]를 클릭
  - AR Default Plane은 인식한 평면(수평/수직)을 시각화하는 AR Plane Mesh Visualizer 스크립트를 포함



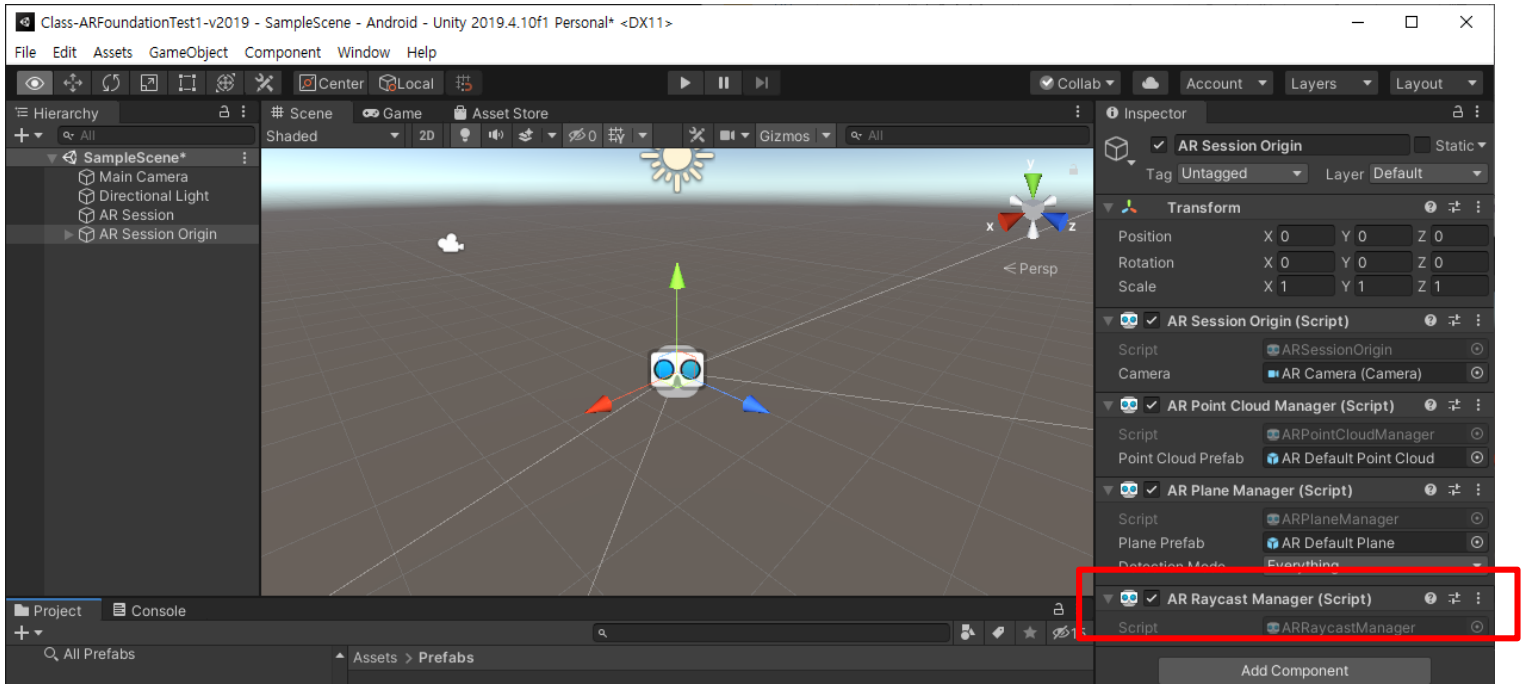
- 하이러리키 뷰에서 AR Default Plane를 프로젝트 뷰로 드래그하여 프리팹으로 만듬
- 프리팹으로 전환한 다음, 하이러리키 뷰에서는 AR Default Plane를 삭제
- AR Session Origin을 선택하고 인스팩터 뷰에서 [Add Component] 버튼을 클릭하여 AR Plane Manager 스크립트를 추가
- 생성한 AR Default Plane 프리팹을 드래그하여 연결



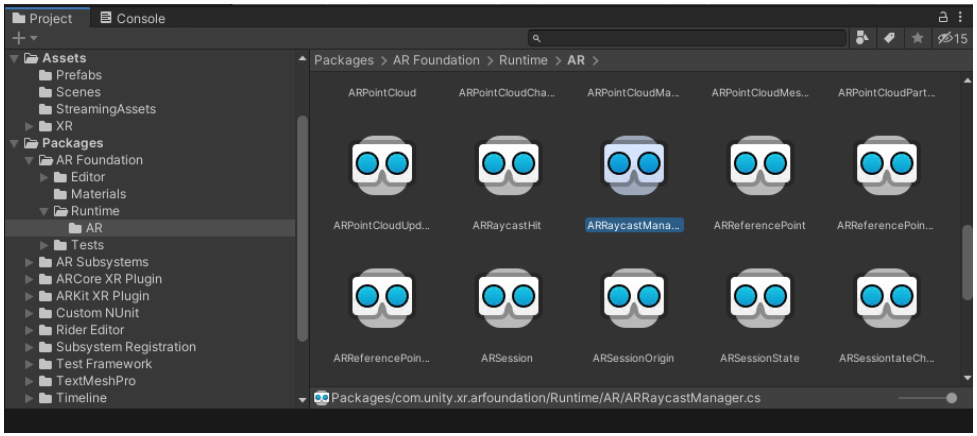
Build and run the app

– 레이캐스팅

- 손으로 터치한 위치에 가상의 물체를 배치 (배치할 모델은 애셋스토어에서 다운로드)
- AR Raycast Manager
  - 하이러리키 뷰에서 AR Session Origin을 선택하고 [Add Component] 버튼을 클릭하여 AR Raycast Manager 스크립트를 추가



– AR Raycast Manager은 현실 세계를 인식하여 생성한 Feature Point 또는 평면 매쉬에 레이를 투사하는 역할을 함



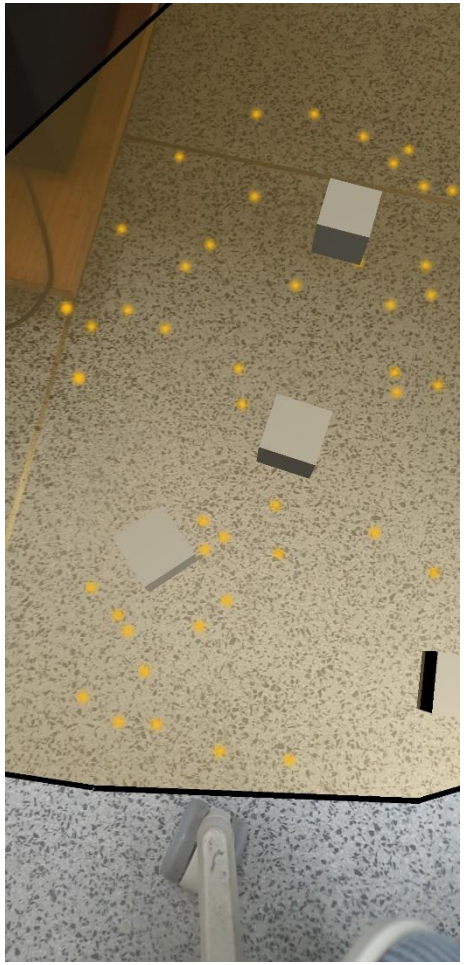
– AR Raycast Manager에서 제공하는 Raycast 메소드의 원형

```
public bool Raycast(
    Vector2 screenPoint,                                // 스크린 좌표
    List<ARRaycastHit> hitResults,                      // 결과값 리스트
    TrackableType trackableTypes = TrackableType.All)  // 검출 범위

public bool Raycast(
    Ray ray,
    List<ARRaycastHit> hitResults,
    TrackableType trackableTypes = TrackableType.All)
```

– TouchManager 스크립트를 생성하고 AR Session Origin 객체로 드래그하여 연결

```
using System.Collections;
using System.Collections.Generic;
using System.Collections.Specialized;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;

public class TouchManager : MonoBehaviour
{
    //생성할 객체
    private GameObject placeObject;
    private ARRaycastManager raycastMgr;
    private List<ARRaycastHit> hits = new List<ARRaycastHit>();

    // Start is called before the first frame update
    void Start()
    {
        // 생성할 큐브를 할당
        placeObject = GameObject.CreatePrimitive(PrimitiveType.Cube);

        // 큐브의 크기를 설정
        placeObject.transform.localScale = Vector3.one * 0.05f;

        // AR Raycast Manager 추출
        raycastMgr = GetComponent<ARRaycastManager>();
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.touchCount == 0) return;
        Touch touch = Input.GetTouch(0);
        if (touch.phase == TouchPhase.Began)
        {
            // 평면으로 인식한 곳만 레이로 검출
            if (raycastMgr.Raycast(touch.position, hits, TrackableType.PlaneWithinPolygon))
            {
                Instantiate(placeObject, hits[0].pose.position, hits[0].pose.rotation);
            }
        }
    }
}
```

Build and run the app



Practice

- 탭을 통해 생성된 객체를 드래그하여 위치를 이동하도록 수정 보완
- Asset Store (또는 Free3d.com)에서 모델링된 객체를 다운로드 받아, (앞의 예제에서) 생성되는 큐브를 대체하도록 수정 보완

– 객체 이동 스크립트 : update() 부분만

```
void Update() {
        if (Input.touchCount == 0) return;
        Touch touch = Input.GetTouch(0);

        //터치 시작시
        if (touch.phase == TouchPhase.Began) {

            Ray ray;
            RaycastHit hitobj;

            ray = arCamera.ScreenPointToRay(touch.position);

            //Ray를 통한 오브젝트 인식
            if(Physics.Raycast(ray, out hitobj))
            {
                //터치한 곳에 오브젝트 이름이 Cube를 포함하면
                if (hitobj.collider.name.Contains("Chicken"))
                {
                    //그 오브젝트를 SelectObj에 놓는다 //터치하고 있는다
                    SelectedObj = hitobj.collider.gameObject;
                    Touched = true;
                }
                //아니면 오브젝트 선택 아닐 시 생성
                else
                {
                    if (raycastMgr.Raycast(touch.position, hits, TrackableType.PlaneWithinPolygon))
                    {
                        placeObject = Instantiate(placeObject, hits[0].pose.position, hits[0].pose.rotation);
                        placeObject.transform.Rotate(0, 180, 0);
                    }
                }
            }
        }
        //터치가 끝나면 터치 끝.
        if(touch.phase == TouchPhase.Ended)
        {
            Touched = false;
        }

        if (raycastMgr.Raycast(touch.position, hits, TrackableType.PlaneWithinPolygon))
        {
            //터치 시 해당 오브젝트 위치 초기화
            if (Touched)
            {
                SelectedObj.transform.position = hits[0].pose.position;
            }
        }
}
```

과제

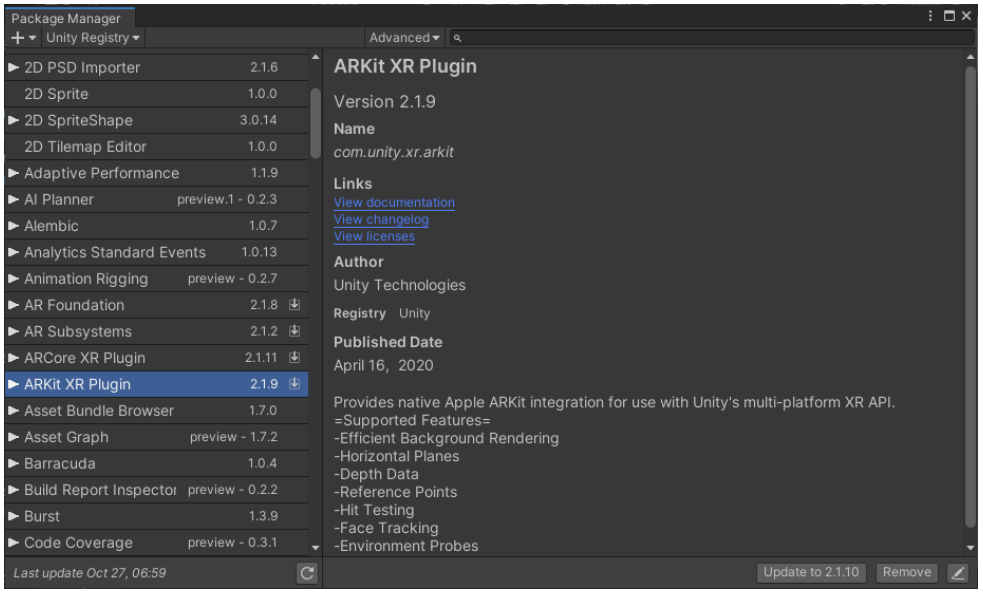
- 얼굴 인식 기능인 Face Tracking 기능을 구현해 봅니다

- Face Tracking
  - 얼굴 인식 기능으로 Face tracking를 구현

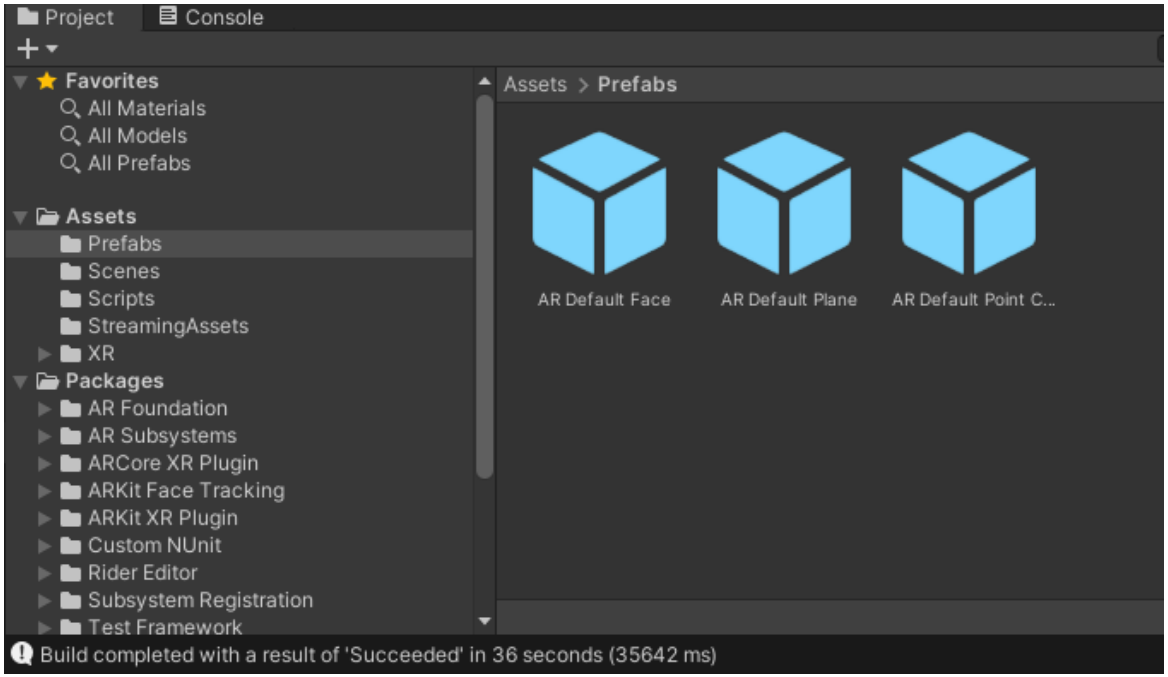
    ARCore는 단순하게 얼굴을 인식하는 기능만을 제공하는 반면에, ARHit는 얼굴인식과 블랜드 셰이프(Blend Shape) 가능을 사용하여 얼굴 근육의 특징을 추적하면서 다양한 표준까지도 표현

    : 블랜드 셰이프 기능은 트루뎁스(True Depth) 카메라가 장착된 아이폰 X이상

- Package Manager에서 ARKit XR Plugin과 ARKit Face Tracking 패키지를 설치

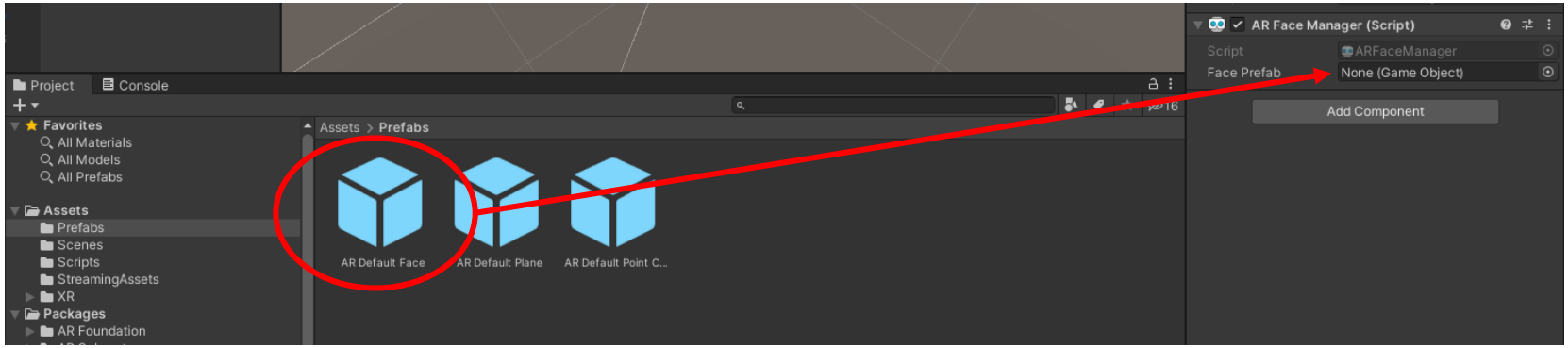


- [GameObject – XR – AR Default Face]를 클릭하여 하이러리키 뷰에 AR Default Face를 추가
- 추가한 AR Default Face를 선택하고 프로젝트 창으로 드래그하여 프리팹으로 만듬
- 프리팹으로 만든 다음, 하이러리키 뷰에서 AR Default Face는 반드시 삭제



– 하이러리키 뷰에서 AR Session Origin을 선택하고 인스팩터 뷰에서 [Add Component] 버튼을 클릭하여 AR Face Manager를 추가

– AR Face Manager의 Face Prefab 속성에 AR Default Face 프리팹을 드래그하여 연결



Build and run the app

- 자신의 얼굴을 비추면 얼굴을 분석하여 메시로 얼굴 마스크를 생성

Class-ARFoundation1-v2019.zip

Download and run the app

- https://github.com/Unity-Technologies/arfoundation-samples

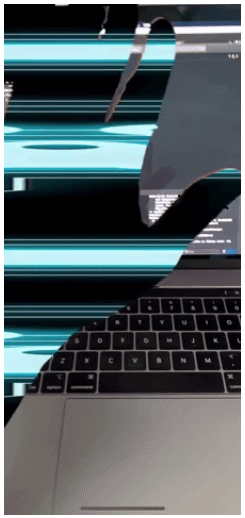
arfoundation-samples-latest-preview.zip

Download and run the app

- https://github.com/dilmerv/UnityARFoundationEssentials

Face Tracking
Generating Masks

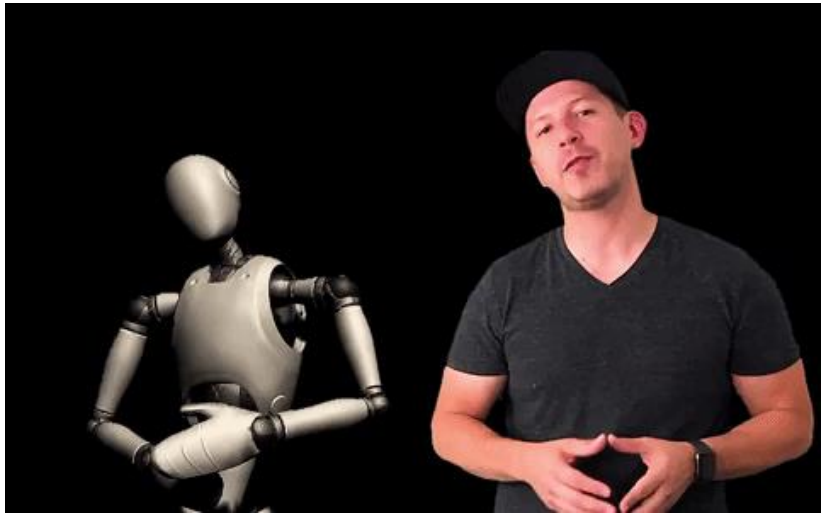
People Occlusion



Body Tracking
w/Fire Particles

.zip

Body Tracking
w/Offset Options



Body Tracking
w/Skeleton made of Cubes
- No Hands

- Full Body

Image Tracking w/Reference Image Name

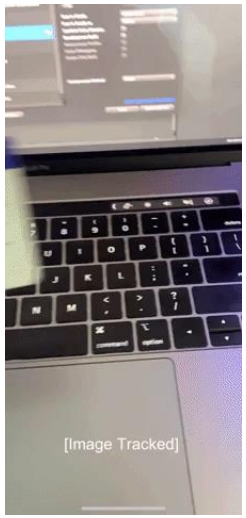
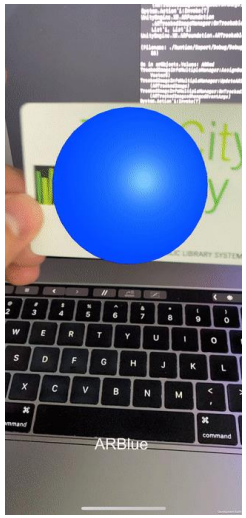
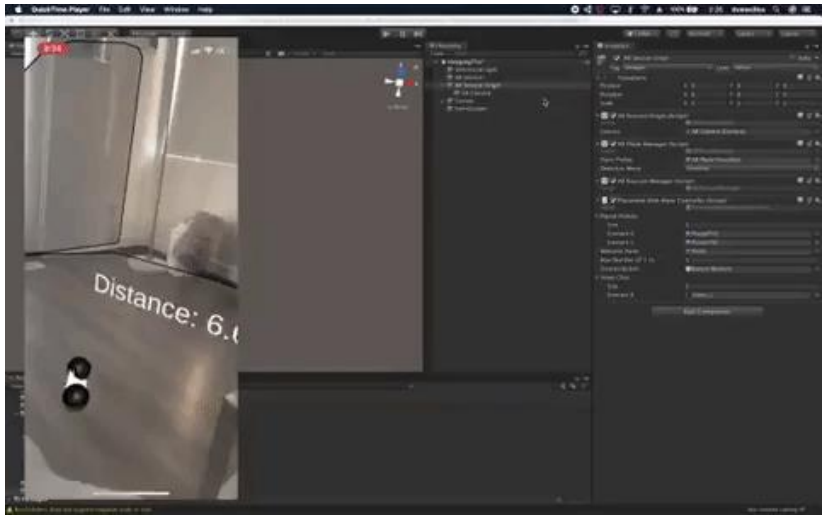


Image Tracking w/A Prefab Per Image Tracked



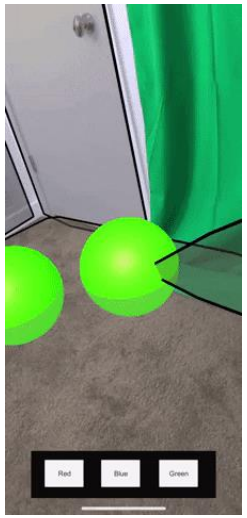
AR Measuring Tape



AR Object Selection

TV Placement on Walls

AR Basic Inventory



AR Realistic Statue