

Capstone Design

Team 12

20165729	Park Sangwoo
20161344	Heo JeongWoo
20161090	Lee Chaemin

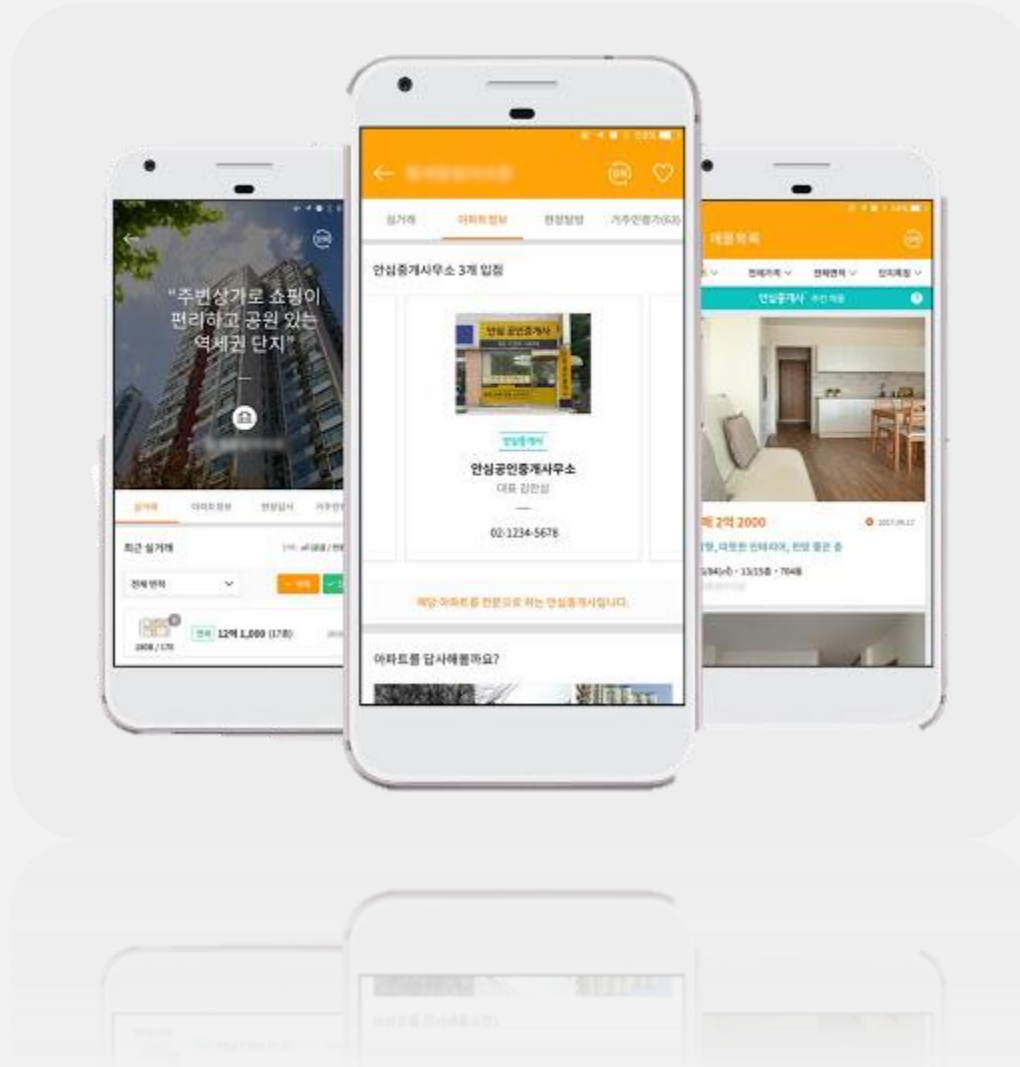
Contents

1. Project Summary
2. Feedback
3. Schedule Modification
4. Progress Report
5. Q&A

















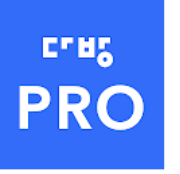







1.

Project Summary

Similar Application



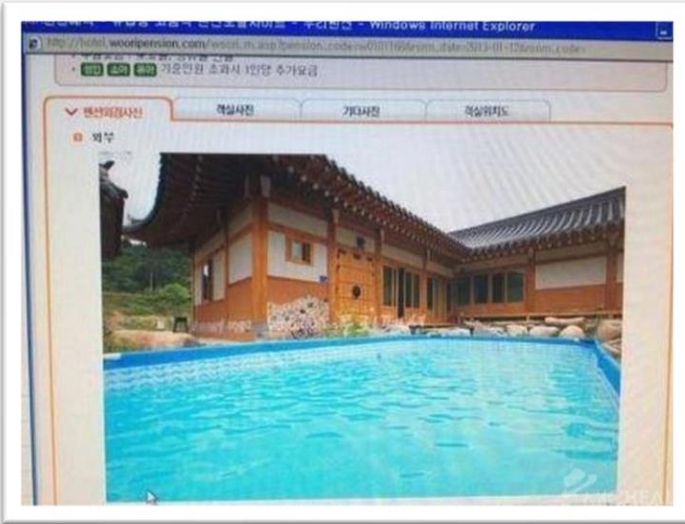
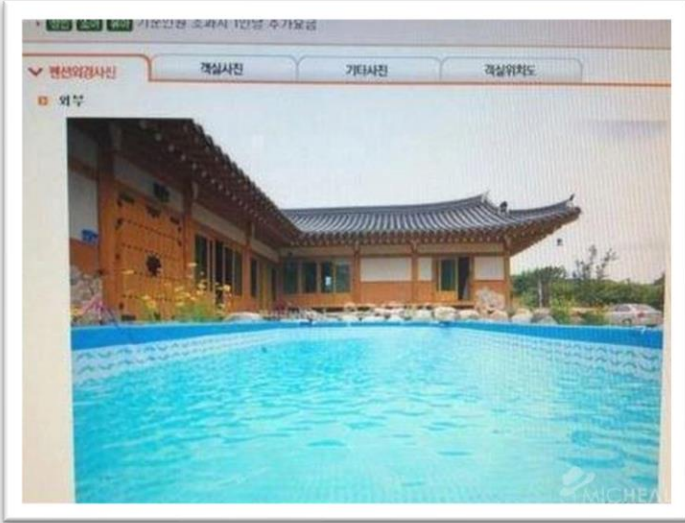
Similar Application

 <p>직방 - No.1 부동산 ZIGBANG</p> <p>★★★★★</p>	 <p>다방 - 대한민국 대표 STATION3</p> <p>★★★★★</p>	 <p>오늘의집 - 1000만인 Bucketplace Inc.</p> <p>★★★★★</p>	 <p>피터팬의 좋은방 구 피터팬의 좋은방 구하기</p> <p>★★★★★</p>	 <p>구해줘홈즈 - 모두의 Nextmobile</p> <p>★★★★★</p>	 <p>집토스부동산 - 거품 집토스</p> <p>★★★★★</p>	 <p>다음 부동산 - 아파트, ZIGBANG</p> <p>★★★★★</p>	 <p>네이버 부동산 - 아파 NAVER Corp.</p> <p>★★★★★</p>	 <p>집나와 - 무료매물등 CACcompany</p> <p>★★★★★</p>
 <p>고방 - 고시원, 쉼어하 2030 맞춤 부동산 서비스</p> <p>★★★★★</p>	 <p>울산직방 리얼소프트_동지</p> <p>★★★★★</p>	 <p>LH임대분양정보 WA Soft</p> <p>★★★★★</p>	 <p>호갱노노 - 아파트 실 호갱노노</p> <p>★★★★★</p>	 <p>부산직방 명문부동산</p> <p>★★★★★</p>	 <p>방픽 - 수많은 부동산 myBangPick</p> <p>★★★★★</p>	 <p>죽방 - 대구부동산 N jookbang</p> <p>★★★★★</p>	 <p>덕방 - 부동산 넘버1 리얼소프트_동지</p> <p>★★★★★</p>	 <p>다방프로 - 부동산 관 STATION3</p> <p>★★★★★</p>
 <p>아파트 실거래가 (아 아실</p> <p>★★★★★</p>	 <p>LH 임대주택, 분양주 Biglove</p> <p>★★★★★</p>	 <p>방굴 - 원룸, 투룸, 오 부동산114</p> <p>★★★★★</p>	 <p>부동산114 부동산114</p> <p>★★★★★</p>	 <p>구방 - 중개사가 알아 구방</p> <p>★★★★★</p>	 <p>교차로 - 부동산, 취 (주)인터넷교차로</p> <p>★★★★★</p>	 <p>한방 - 부동산,매물,시 [한국공인중개사협회]</p> <p>★★★★★</p>	 <p>다룸-원룸,투룸,하메, XENGROUP</p> <p>★★★★★</p>	 <p>공짜방 Smart Corp.</p> <p>★★★★★</p>

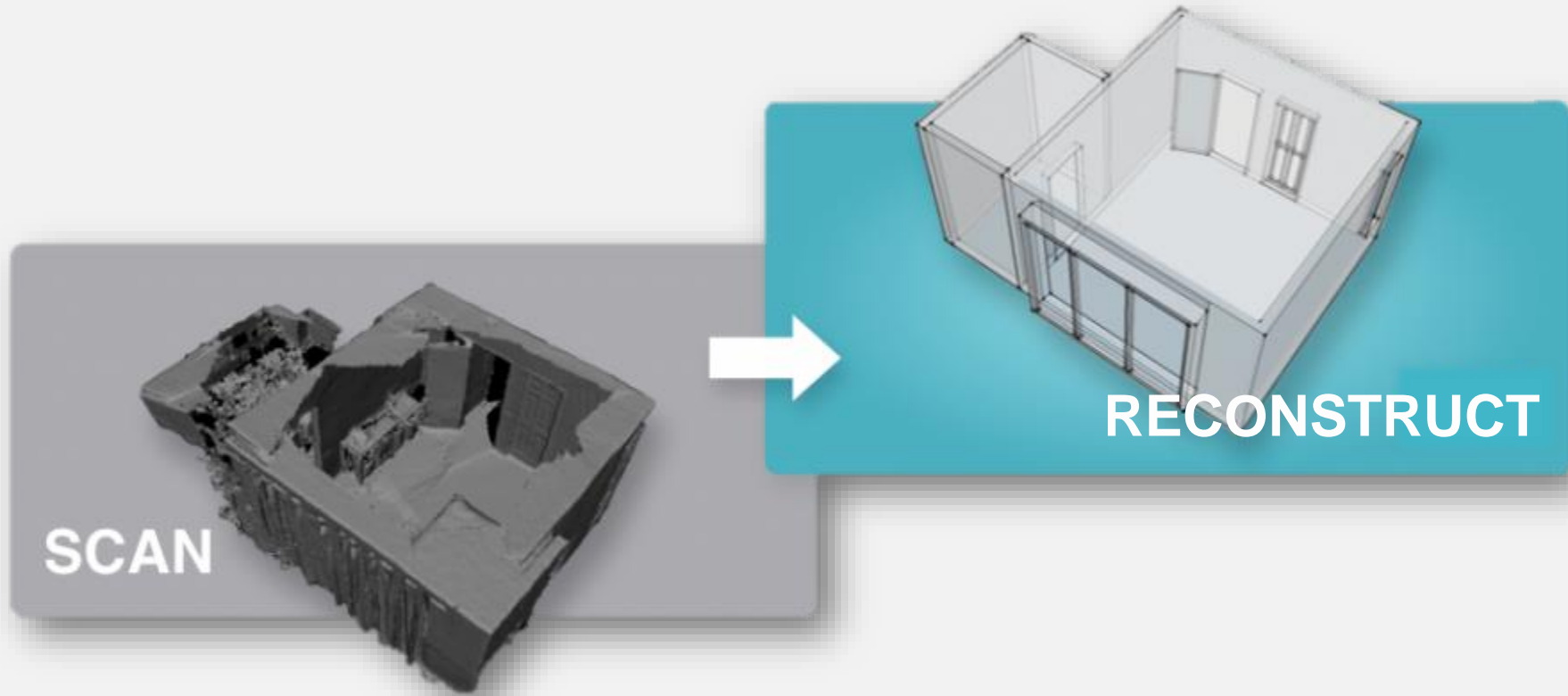
A lot of Real Estate Brokerage Applications are already existing



Similar Application



Reconstruct



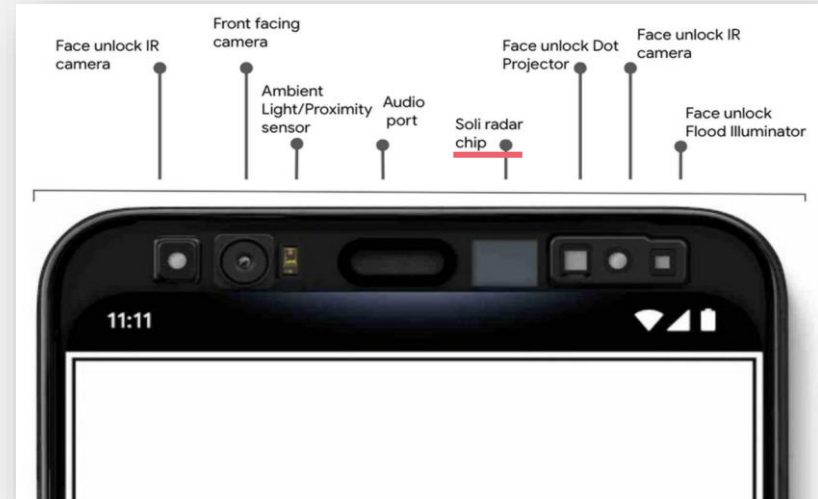
Using 3D Scanning!



With ToF, Lidar sensors.

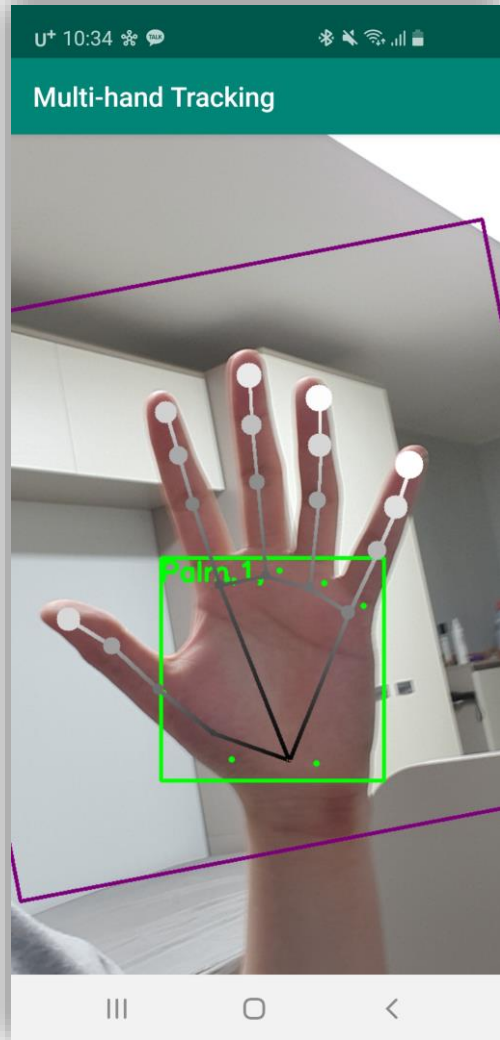


3D Scanning



VR Compatible (Optional)





Control by Hand Tracking

Not a just virtually place furnishings application



2.

Feedback

What's differences?

What are the differences between other applications, such as IKEA applications?

What's differences?

SWOT Research

Strength

- You can check the apartment sales reviews in the application
Categorized according to age, gender, rent-based, etc.
- Convenience and Speed of Real Estate Exploration
24-hour service via the Internet.
You can easily check for sale without having to go to a brokerage house.
Explore things beyond time and space with real-time feedback.

Weakness

- There are many false offerings.
- Difficult to know the exact information about the house.

Opportunity

- An increase in the share of monthly rent, and leaseholds.
- Real estate agent's interest
Meditation through mobile services can save both tenants and brokers time and money in real estate sales transactions.

Threat

- Multiple Competitors increase.



3.

Schedule Modification

Total Schedule

	September				October				November				December		
	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16
Development Contents															
Content Acquisition & Data Survey															
Study android API for camera															
Implementing 3D reconstruction using unity															
Implementing RGB-D camera scanning															
Implementing application UI															
Study Unity3D and AR Core															
Implementing Sensor tracking															
Preparing Midterm Demonstration and Presentation															
Test and Debugging															
Preparing Final Demonstration and Presentation															
Writing manuals and reports															

1. Heo Jeong-woo

	September				October				November				December		
Development Contents	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16
Content Acquisition & Data Survey															
Study android API for camera															
Implementing RGB-D camera scanning															
Implementing application UI															
Implementing Sensor tracking															
Preparing Midterm Demonstration and Presentation															
Test and Debugging															
Preparing Final Demonstration and Presentation															
Writing manuals and reports															

2. Lee Chae-min

	September				October				November				December		
Development Contents	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16
Content Acquisition & Data Survey															
Study android API for camera															
Implementing 3D reconstruction using unity															
Implementing application UI															
Study Unity3D and AR Core															
Preparing Midterm Demonstration and Presentation															
Test and Debugging															
Preparing Final Demonstration and Presentation															
Writing manuals and reports															

3. Park Sang-woo

	September				October				November				December		
Development Contents	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16
Content Acquisition & Data Survey															
Study android API for camera															
Implementing 3D reconstruction using unity															
Implementing application UI															
Study Unity3D and AR Core															
Preparing Midterm Demonstration and Presentation															
Test and Debugging															
Preparing Final Demonstration and Presentation															
Writing manuals and reports															

4.

Progress Report

Android API for Multi-Camera

REQUEST_AVAILABLE_CAPABILITIES_LOGICAL_MULTI_CAMERA

Added in API level 28

```
public static final int REQUEST_AVAILABLE_CAPABILITIES_LOGICAL_MULTI_CAMERA
```

The camera device is a logical camera backed by two or more physical cameras.

In API level 28, the physical cameras must also be exposed to the application via `CameraManager.getCameraIdList()`.

Starting from API level 29:

- Some or all physical cameras may not be independently exposed to the application, in which case the physical camera IDs will not be available in `CameraManager.getCameraIdList()`. But the application can still query the physical cameras' characteristics by calling `CameraManager.getCameraCharacteristics(String)`.
- If a physical camera is hidden from camera ID list, the mandatory stream combinations for that physical camera must be supported through the logical camera using physical streams. One exception is that in API level 30, a physical camera may become unavailable via `CameraManager.AvailabilityCallback.onPhysicalCameraUnavailable` callback.

Combinations of logical and physical streams, or physical streams from different physical cameras are not guaranteed. However, if the camera device supports `CameraDevice.isSessionConfigurationSupported`, application must be able to query whether a stream combination involving physical streams is supported by calling `CameraDevice.isSessionConfigurationSupported`.

Camera application shouldn't assume that there are at most 1 rear camera and 1 front camera in the system. For an application that switches between front and back cameras, the recommendation is to switch between the first rear

OR



ARCore

Android Multi-Camera Camera2 API

Android API for Multi-Camera

app / com.dhirajgupta.multicam.services / ManagedCamera

ManagedCamera

```
class ManagedCamera
```

This class is heavily inspired from Google's Camera2Basic sample at <https://github.com/googlesamples/android-Camera2Basic>. Heavy refactoring has been carried out to move all the code from the Fragment in the sample to a self contained [ManagedCamera](#) class that can take care of its own threads and is almost fully self sufficient. By modularizing the Camera in this way we are able to easily create multiple instances (two in this case) while the implementation code remains common between both the cameras.

Constructors

Name	Summary
<init>	<pre>ManagedCamera(systemId: String, threadName: String, textureView: AutoFitTextureView, listener: ManagedCameraStatus)</pre> <p>This class is heavily inspired from Google's Camera2Basic sample at https://github.com/googlesamples/android-Camera2Basic. Heavy refactoring has been carried out to move all the code from the Fragment in the sample to a self contained ManagedCamera class that can take care of its own threads and is almost fully self sufficient. By modularizing the Camera in this way we are able to easily create multiple instances (two in this case) while the implementation code remains common between both the cameras.</p>

Properties

Name	Summary
activity	<pre>val activity: Activity</pre>
cameraState	<pre>var cameraState: Int</pre> The current state of camera state for taking pictures. REFACTORING NOTE: CAMERA_STATE_IDLE has been

Android Camera2Basic Sample

This sample demonstrates using the Camera2 API to capture a JPEG, DEPTH or RAW frame. Check the source code to see a simple example of how to display the camera preview and capture a still image using the default configuration with the selected pixel format.

Introduction

The [Camera2 API](#) allows users to capture RAW images, i.e. unprocessed pixel data directly from the camera sensor that has not yet been converted into a format and colorspace typically used for displaying and storing images viewed by humans. The [DngCreator](#) class is provided as part of the Camera2 API as a utility for saving RAW images as DNG files.

This sample displays a live camera preview in a TextureView, and saves JPEG and DNG file for each image captured.

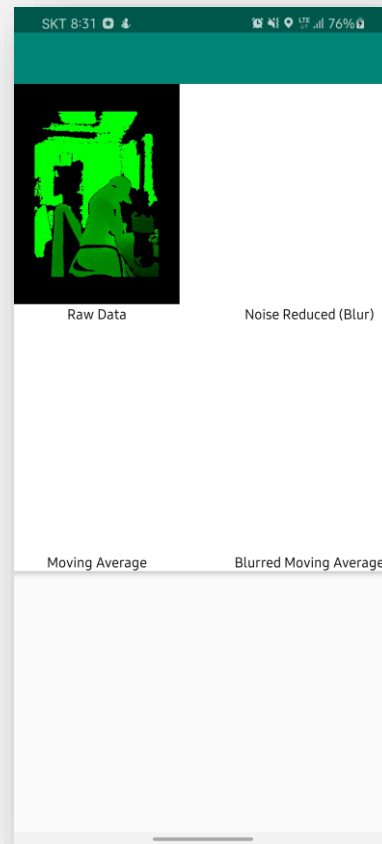
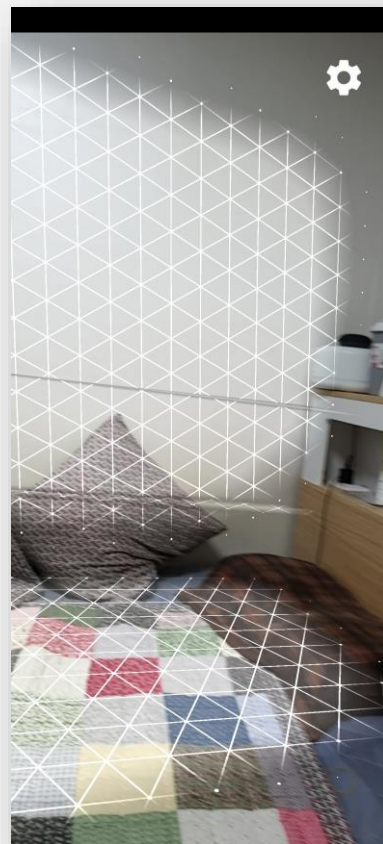
Pre-requisites

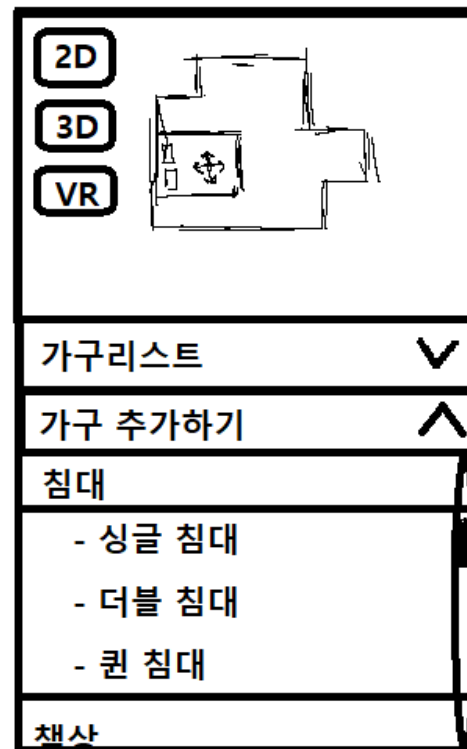
- Android SDK 29+
- Android Studio 3.5+

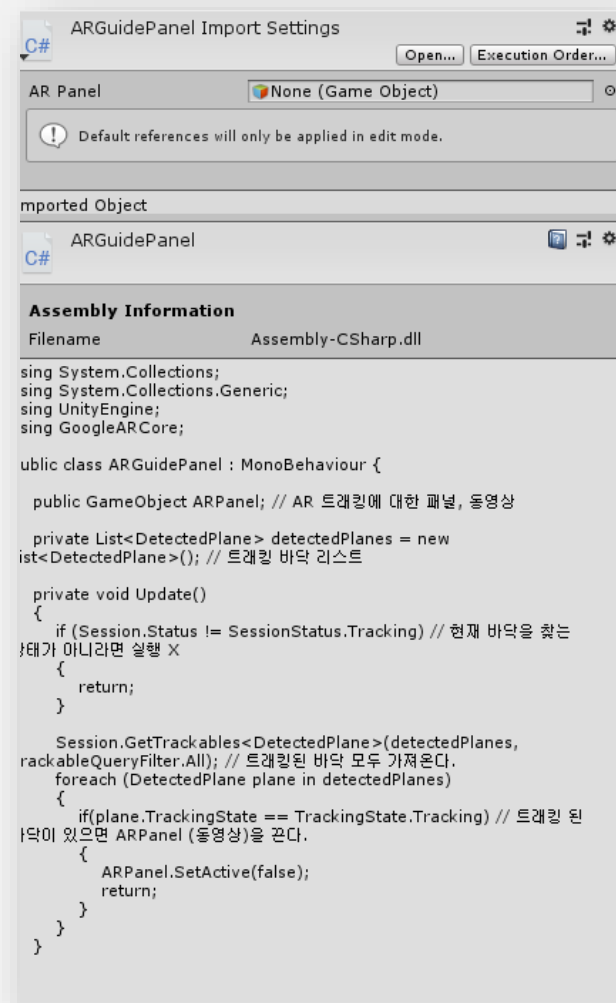
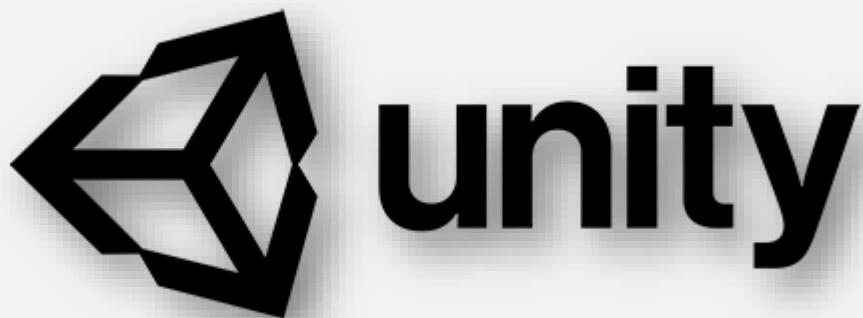
Screenshots



Android API for Multi-Camera







5.

Q&A