

작품명 : 시장 네비게이션

팀명 : 가는 날이 장날

팀번호	16046
대표소속	황금중학교
팀원이름	허채원, 허채영, 김민재
지도교사(멘토) 이름	허준성
시연 동영상(URL)	<a href="https://youtu.be/jH4AbXdKfJ8">https://youtu.be/jH4AbXdKfJ8</a>

# 작품 요약서

## 1. 작품 제목

이미지 방식의 시장 네비게이션

## 2. 작품 개요

재래 시장이 IT에 뒤쳐지고 낙후되어 청소년들이 가기 싫다는 안 좋은 이미지를 개선하고 싶습니다. 재래 시장에서 물건을 사려고 방문하면, 어디에 무엇을 얼마에 파는 지를 시장을 둘러보면서 묻기 전에는 알 수가 없습니다. 특정 상점을 GPS로 찾아가려 해도, 시장이 실내에 위치하거나 실내의 좁은 공간에 가게와 노점상이 밀집해 있어 GPS 방식의 길 찾기가 도움이 되지 않습니다.

우리는 이미지 방식의 시장 네비게이션을 만들어서 내가 찾는 가게가 어디에 있는지 검색하고 가게로 가는 길을 네비게이션 방식으로 안내해주도록 하겠습니다. 장보기 미션 기능은 길 찾기 기능과 보호자의 모니터링 그리고 대화 기능을 이용하여 게임처럼 심부름을 완수하는 기능입니다. 장보기 미션 기능은 어린이나 청소년들이 재래 시장에 가는 것을 재미있는 놀이로 인식하여 한번 가보게 만들어서 거부감을 최대한 줄여 주고 싶습니다.

## 3. 작품 설명 (간단하게 설명)

### 3-1. HW의 구성 및 특징

이미지 방식 시장 네비게이션은 안드로이드 스마트폰에서 별도의 하드웨어 장치없이 구동되는 어플입니다

위치인식을 위하여 스마트폰용 카메라를 사용하므로 반드시 폰에 카메라가 장착되어 있어야 합니다

### 3-2. SW의 구성 및 특징

소프트웨어는 크게 3가지로 구성되어 있습니다

1. 영상을 비교하여 유사한 영상을 찾아내는 모듈
  - OPENCV 라이브러리의 사용
  - 카메라 영상의 이미지 프로세싱 (blur, resize)
  - 영상의 특징 점 추출 및 Matching

- 히스토그램 비교
- 2. 시장 맵 을 렌더링하고 사용자의 위치와 목적지를 표시하는 모듈
  - Canvas 렌더링
  - 사용자 위치의 Animation
- 3. 장보기 미션을 위한 채팅 기능
  - Firebase 모듈의 사용
  - Firebase 모듈을 이용한 DB 기능
  - 장바구니 기능

### 3-3. 기타

개발 일정이 촉박하여 프로그램 개발이 완료된 상태는 아닙니다

현재 70% 개발 완료되었습니다

독립적으로 개발한 모듈을 통합시키는 작업이 남아있습니다



프로그램 구상	
프로젝트 기간 및 목표	
<b>기간</b>	6월 15일 ~ 6월 20일
<b>목표</b>	전체 프로그램의 만들어질 모습을 구상한다
프로젝트 활동내용	
<p>자동차에 있는 네비게이션 같이 만들고 싶었습니다</p> <p>네비게이션이 가져야 할 기능과 모듈을 구상하고 회의하여 결정하였습니다</p> <p>프로젝트는 세가지 모듈로 생각했어요</p> <ul style="list-style-type: none"> <li>(1) 영상 비교하는 것</li> <li>(2) 지도 표시하는 것</li> <li>(3) 지도 보며 채팅 하는 것</li> </ul> <p>시행착오는 많아요</p> <p>모듈별로 나누어서 계획을 잘 세웠어요 그런데 그렇게 완성한 걸 합치는 과정에서 문제가 많이 생겼어요</p> <p>분명 독립적으로 만든 영상인식 부분이 합치면서 에러가 나요. Fragment에서 안되는 기능인데 붙였던 것 같아요. 통합과정이 프로그램 제출 며칠전에 잡혀 있어서 제출전까지 다 못 붙인 부분도 있어요</p> <p>다음 일정에서 통합과정에 시간을 많이 넣겠습니다</p>	



이번 프로젝트는 계획 부분에 시간을 많이 사용한 것 같습니다  
 그런데 테스트 부분에 시간을 할당하지 못해서 매번 일정이 늦어집니다  
 테스트에서 발생한 버그 때문에 처리하지 못한 일정이 여러가지입니다  
 다음 계획에서는 테스트에 많은 시간을 넣겠습니다

**지도교사 확인 및 의견**

많은 도움은 되었지만 아쉽기도 한 부분 이였습니다.

아이들이 조금 해보고 안된다며 네이버 검색으로 다른 방법을 자꾸 찾아서  
 끈기 있게 해결하는 모습이 없어서 아쉬웠습니다

(서명)  
 허준성

## 안드로이드에 OPEN CV 연결하기

### 프로젝트 기간 및 목표

<b>기간</b>	6월 25일 ~ 7월 15일
<b>목표</b>	Android 프로그래밍에 OpenCV 라이브러리를 접목시킨다.

### 프로젝트 활동내용

안드로이드 프로그램은 JAVA로 개발하므로 프로그램의 동작이 느립니다

카메라로 입력된 영상을 실시간으로 처리할 수 있는 방법을 네이버와 구글에서 검색하였습니다

Windows 프로그램의 영상처리에 많이 사용하는 OpenCV를 안드로이드에 JAVA로 적용할 수 있음을 찾았습니다

첫번째 시행착오 -

안드로이드에서 OpenCV를 사용하려면 무조건 NDK를 설치해야 하는 줄 알고 설치 후 C언어로 작성을 하였는데 너무 어려워서 할 수가 없었습니다

다시 찾아보니 JAVA로도 OpenCV를 사용할 수 있음을 알게 되었습니다

그래서 다시 처음부터 JAVA로 엔진을 작성하였습니다

두번째 시행착오 -

OpenCV는 버전이 여러 개 있습니다

안드로이드에서 실행하기 위해서는 폰에도 라이브러리가 설치되어 있어야 합니다

폰에 설치되는 라이브러리는 3.4버전인데 개발에 사용하는 버전과 일치하지 않으면 이상하게 작동됩니다, 이 이유로 3일쯤 고생하였습니다

더 깊이 있게 OpenCV를 사용하려면 NDK를 사용하여 C로 작업을 하라고 하는데 너무 어렵습니다

세번째 시행착오 -

안드로이드 작업의 시행착오입니다

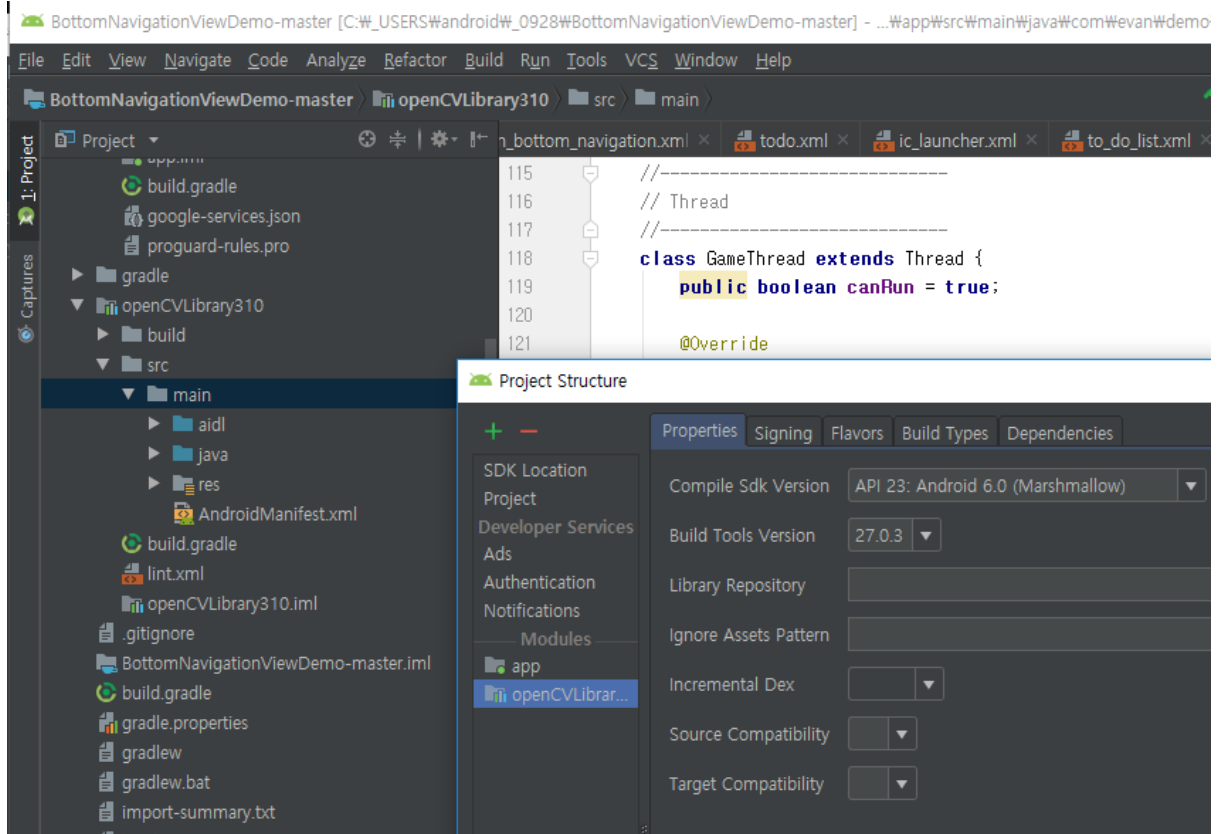
개발을 진행하는 동안 갑자기 프로그램이 계속 죽는 현상이 나타났습니다.

어떤 것을 수정해도 계속 죽었습니다

나중에 알고 보니 안드로이드 프로그램의 보안이 강화되면서 폰에 설치된 시장 네비게이션

어플의 storage permission을 변경하여야 하는 문제였습니다

아래 그림은 OpenCV를 설정하는 모습입니다



- 작업 담당 (허채원)

아래의 코드는 OpenCV를 연결하여 이미지 프로세싱 작업을 진행하는 코드입니다

```
try {
    Imgproc.cvtColor(mRgba2, mRgba2, Imgproc.COLOR_BGR2GRAY);
    Imgproc.warpAffine(mRgba2, mRgba2, rotateM, mRgba2.size() );
    Imgproc.resize( mRgba2, mRgba2, sz );
    //Size sz2 = new Size(3,3 );
    //Imgproc.blur( mRgba2, mRgba2,sz2);
    mRgba2.convertTo(mRgba2, 0); //converting the image to match with the type of the
    cameras image
    bitmap2 = Bitmap.createBitmap(mRgba2.cols(), mRgba2.rows(), Bitmap.Config.ARGB_8888);

    mRgba2.copyTo(mRgba22);
    bitmap22 = Bitmap.createBitmap(mRgba22.cols(), mRgba22.rows(),
    Bitmap.Config.ARGB_8888);

    //issue - this is going to create a Runnable for each frame
    Runnable runnable = new Runnable() {
        @Override
```

```

public void run() {
    try {
        Utils.matToBitmap(mRgba2, bitmap2 );
        iview11.setImageBitmap( bitmap2 );
        iview11.invalidate();
        sleep(1);

        // Compare2Image();
    } catch (Exception e) {
        //
    }
}
};
new Thread(runnable).start();
}
catch(Exception ex) {
    Log.i(TAG, "Exception onCameraFrame");
}
}

```

**이번 프로젝트에 대한 반성 / 다음 프로젝트 계획**

안드로이드도 능숙하게 사용 못하는데 OpenCV까지 같이 하려니 너무 힘들었습니다  
 다음에는 OpenCV를 더 공부하여 전문가처럼 되고 싶습니다

**지도교사 확인 및 의견**

<p>가장 핵심이 되는 부분인데 한글로 된 자료도 부족하고 예제도 부족하여 아이들이 힘들어 했습니다. 그러나 적용해서 나오는 결과를 보고는 모두 만족하였습니다</p>	<p>(서명) 허준성</p>
--	---------------------

## 영상 비교 알고리즘의 구현

### 프로젝트 기간 및 목표

기간	7월 20일 ~ 8월 20일
목표	두 영상을 비교하여 동일한 영상인지를 판단한다

### 프로젝트 활동내용

Open CV 알고리즘에 의한 동일 이미지 판독 알고리즘을 구성하는게 목표였습니다

특징점에 의한 이미지 매칭 방법을 참고자료에 따라 구현하였습니다

실제 테스트에서 잘 찾는 경우와 찾기 어려운 부분이 발생하였습니다

특징점 매칭 방법에 히스토그램 비교 방법을 추가하여 더 정확하게 비교하도록 하였습니다

```
public void compare2() {
    if (descriptor == DescriptorExtractor.BRIEF)
        descriptorType = "BRIEF";
    else if (descriptor == DescriptorExtractor.BRISK)
        descriptorType = "BRISK";
    else if (descriptor == DescriptorExtractor.FREAK)
        descriptorType = "FREAK";
    else if (descriptor == DescriptorExtractor.ORB)
        descriptorType = "ORB";
    else if (descriptor == DescriptorExtractor.SIFT)
        descriptorType = "SIFT";
    else if (descriptor == DescriptorExtractor.SURF)
        descriptorType = "SURF";

    descriptorType = "BRISK";

    try {
        startTime = System.currentTimeMillis();

        bmpimg1 = bmpimg1.copy(Bitmap.Config.ARGB_8888, true);
        bmpimg2 = bmpimg2.copy(Bitmap.Config.ARGB_8888, true);

        Toast.makeText(this, "compare2--1", Toast.LENGTH_SHORT).show();

        img1 = new Mat();
        img2 = new Mat();
        Utils.bitmapToMat(bmpimg1, img1);
        Utils.bitmapToMat(bmpimg2, img2);
        Imgproc.cvtColor(img1, img1, Imgproc.COLOR_BGR2RGB);
        Imgproc.cvtColor(img2, img2, Imgproc.COLOR_BGR2RGB);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```

Size sz = new Size(256,256);
Imgproc.resize( img1, img1, sz );
Imgproc.resize( img2, img2, sz );

Log.d("log", "img1:" + img1.rows() + "x" + img1.cols() + " img2:" + img2.rows() +
"x" + img2.cols());
Mat hist1 = new Mat();
Mat hist2 = new Mat();
Log.d("log", "Calc 8");

MatOfInt histSize = new MatOfInt(180);
MatOfInt channels = new MatOfInt(0);
ArrayList<Mat> bgr_planes1 = new ArrayList<Mat>();
ArrayList<Mat> bgr_planes2 = new ArrayList<Mat>();
Log.d("log", "Calc 9");

Core.split( img1, bgr_planes1);
Core.split( img2, bgr_planes2);
MatOfFloat histRanges = new MatOfFloat(0f, 180f);
boolean accumulate = false;
Imgproc.calcHist(bgr_planes1, channels, new Mat(), hist1, histSize, histRanges,
accumulate);
Core.normalize(hist1, hist1, 0, hist1.rows(), Core.NORM_MINMAX, -1, new Mat());
Imgproc.calcHist(bgr_planes2, channels, new Mat(), hist2, histSize, histRanges,
accumulate);
Core.normalize(hist2, hist2, 0, hist2.rows(), Core.NORM_MINMAX, -1, new Mat());

hist1.convertTo(hist1, CvType.CV_32F);
hist2.convertTo(hist2, CvType.CV_32F);

Log.d("log", "Calc 10");

compare = Imgproc.compareHist(hist1, hist2, Imgproc.CV_COMP_CHISQR);

Log.d("log", "compare: " + compare);
if (compare > 0 && compare < 1500) {
    Toast.makeText(this, "Images may be possible duplicates, verifying",
Toast.LENGTH_LONG).show();
    //new AsyncTask(MainActivity.this).execute();
} else if (compare == 0)
    Toast.makeText(this, "Images are exact duplicates", Toast.LENGTH_LONG).show();
else
    Toast.makeText(this, "Images are not duplicates", Toast.LENGTH_LONG).show();

Toast.makeText(this, "compare2—2", Toast.LENGTH_SHORT).show();

```

```

    detector = FeatureDetector.create(FeatureDetector.PYRAMID_FAST);
    DescExtractor = DescriptorExtractor.create(descriptor);
    matcher = DescriptorMatcher
        .create(DescriptorMatcher.BRUTEFORCE_HAMMING);

    keypoints = new MatOfKeyPoint();
    dupKeypoints = new MatOfKeyPoint();
    descriptors = new Mat();
    dupDescriptors = new Mat();
    matches = new MatOfDMatch();
    detector.detect(img1, keypoints);
    Log.d("LOG!", "number of query Keypoints= " + keypoints.size());
    detector.detect(img2, dupKeypoints);
    Log.d("LOG!", "number of dup Keypoints= " + dupKeypoints.size());
    // Describe keypoints
    DescExtractor.compute(img1, keypoints, descriptors);
    DescExtractor.compute(img2, dupKeypoints, dupDescriptors);
    Log.d("LOG!", "number of descriptors= " + descriptors.size());
    Log.d("LOG!",
        "number of dupDescriptors= " + dupDescriptors.size());

    Toast.makeText(this, "compare2—3", Toast.LENGTH_SHORT).show();

    // matching descriptors
    matcher.match(descriptors, dupDescriptors, matches);
    Log.d("LOG!", "Matches Size " + matches.size());
    // New method of finding best matches
    List<DMatch> matchesList = matches.toList();
    List<DMatch> matches_final = new ArrayList<DMatch>();
    for (int i = 0; i < matchesList.size(); i++) {
        // Log.d("LOG!", i + " Matches dist -> " + matchesList.get(i).distance +
        "mindist : " + min_dist);
        if (matchesList.get(i).distance <= min_dist) {
            matches_final.add(matches.toList().get(i));
        }
    }

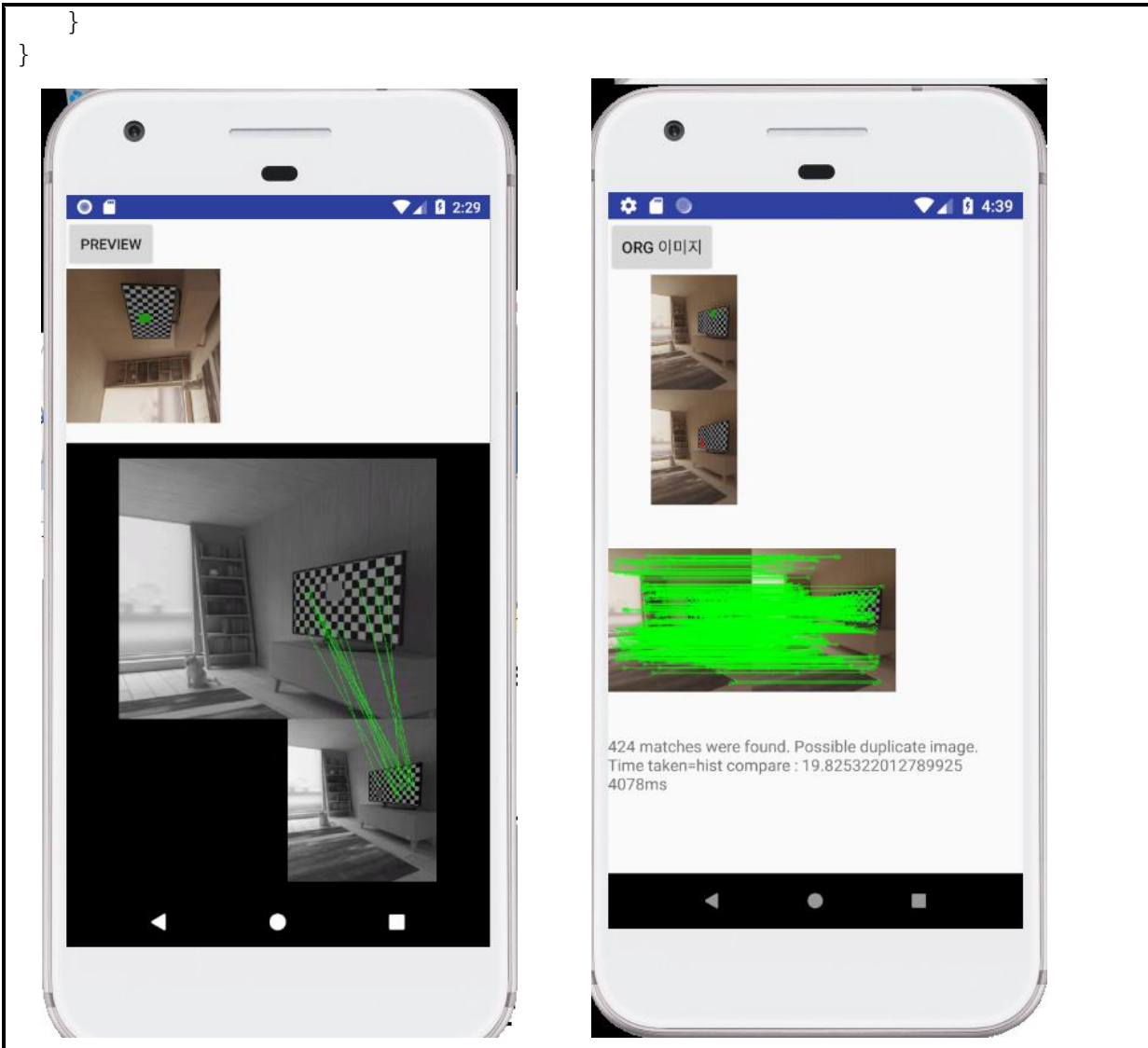
    matches_final_mat = new MatOfDMatch();
    matches_final_mat.fromList(matches_final);

    Toast.makeText(this, "compare2—4", Toast.LENGTH_SHORT).show();

    List<DMatch> finalMatchesList = matches_final_mat.toList();
    final int matchesFound = finalMatchesList.size();
    Log.d("LOG!", "Matches Final Size " + matches.size());

    Toast.makeText(this, "compare2—end", Toast.LENGTH_SHORT).show();
} catch (Exception e) {
    e.printStackTrace();

```



왼쪽 그림은 실시간 영상 비교 화면입니다.

영상이 실시간 입력될 때 한장 한장 따로 비교는 너무 민감한 것 같습니다

두장씩 합친 평균 값으로 영상을 비교하면 좀 더 안정적일 것 같습니다

주 작업진행자 허채원

작업 지도 허준성

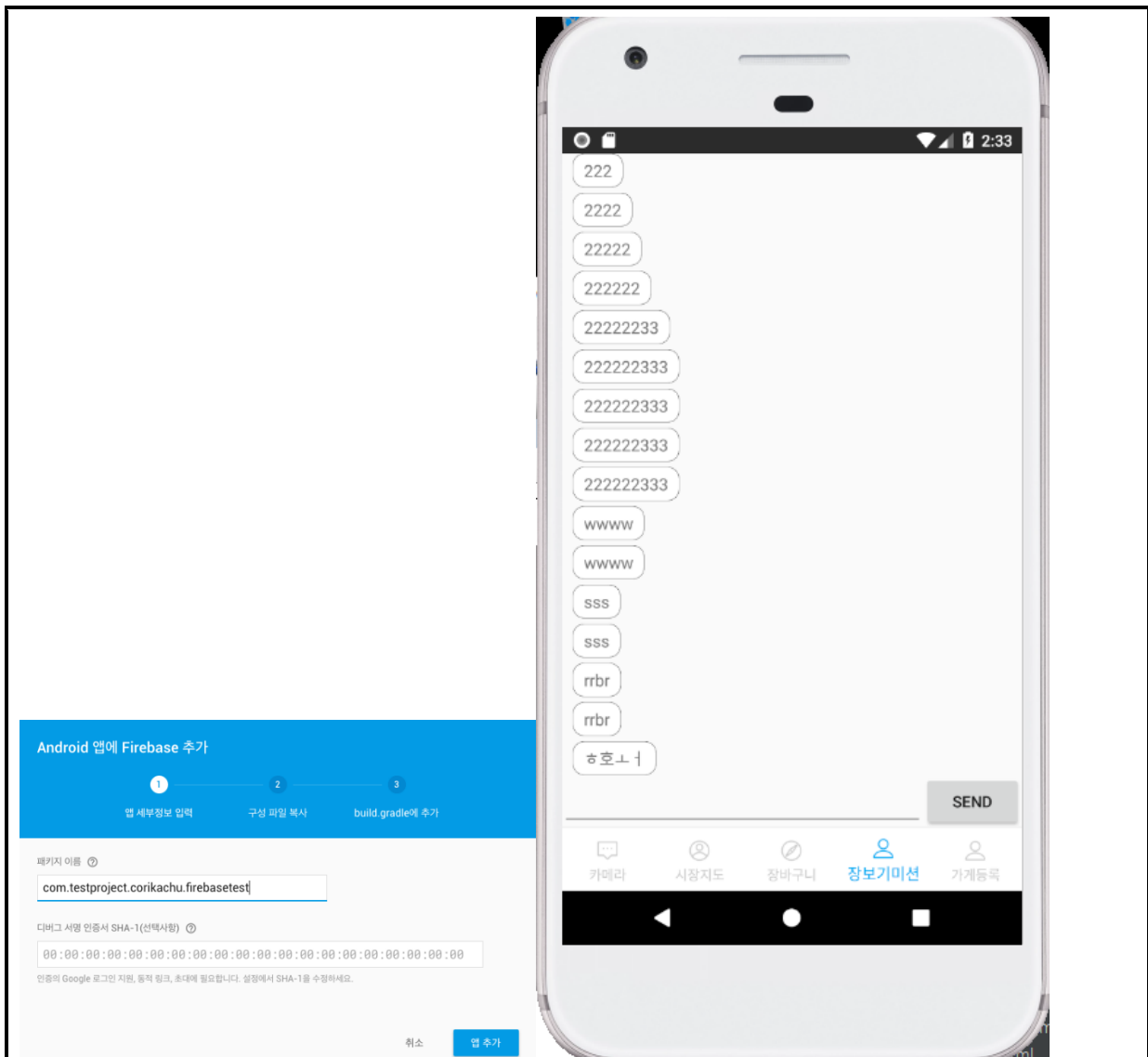
이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

실시간 비교를 위해서 미리 입력된 영상은 특징점을 먼저 계산해 두어야 겠습니다  
 지금 계산량이 많은 지 30분 이상 사용하면 폰이 뜨거워집니다

**지도교사 확인 및 의견**

<p>OpenCV에 있는 특징점 추출 함수와 매칭 함수를 다양하게 조합하여 사용해 보았습니다. 생각보다 결과가 잘 나왔습니다.</p> <p>실시간 보다는 타이머를 사용하여 일정 간격이나 걸음이 멈출 때마다 계산 하는게 배터리 타임에도 좋을 것 같습니다</p>	<p>(서명)</p> <p>허준성</p>
--	------------------------

채팅기능	
프로젝트 기간 및 목표	
<b>기간</b>	6월 25일 ~ 8월 25일
<b>목표</b>	장보기 미션에서 보호자와의 채팅 기능을 구현합니다
프로젝트 활동내용	
<p>장보기 미션에서 보호자와의 미션에 대한 이야기와 위치정보를 나눌 수 있는 채팅 기능을 만듭니다</p> <p>“문과생도 만드는 채팅”이라는 유튜브 강의를 참조하여 개발하였습니다</p> <p>Firebase를 처음 사용하는데 설정하는 부분이 너무 많아서 힘들었습니다.</p> <p>아직 사용자 id 등록하는 부분의 구현이 완전하게 되지 않았습니다.</p> <p>사용자 인증이 완료되면 상대방에게 채팅을 할 수 있습니다.</p> <p>지금은 모두와 같이 하는 게시판같이 사용되고 있습니다.</p> <p>다음은 인터넷에서 본 Firebase Realtime Database의 장점입니다.</p> <p>(1) cloud-hosted database의 사용으로 개발 시간 단축</p> <p>database가 실시간으로 연동된다는 것은 모바일 개발에 있어서 backend개발에 조금은 덜 집중해도 괜찮다는 이야기입니다. 따로 서버를 구축하지 않고 데이터베이스를 사용할 수 있다는 이점은 상당한 장점으로 다가옵니다.</p> <p>(2) Offline을 대비한 내부 캐시</p> <p>여기서는 Offline을 위한 설정을 다루지는 않았지만, 몇몇 설정을 통해 네트워크 연결이 불안정했을 때를 대비해서 내부 캐시에 데이터를 저장할 수 있습니다. 연결이 끊겼을때 수정된 데이터들을 큐에 저장해 두고 다시 연결 되었을 때 동기화합니다.</p> <p>다음은 Firebird 홈페이지에서 사용자 프로그램을 등록하고 계정을 만드는 모습입니다.</p>	



## Realtime Database 사용해서 채팅 보내기

```
sendButton.setOnClickListener((view) -> {
    ChatData chatData = new ChatData(userName,
editText.getText().toString()); // 유저 이름과 메시지로 chatData 만
들기
    databaseReference.child("message").push().setValue(chatData);
// 기본 database 하위 message 라는 child 에 chatData 를 list 로 만들기
    editText.setText("");
});
```

## Realtime Database 사용해서 채팅 받기

```

databaseReference.child("message").addChildEventListener(new
ChildEventListener() { // message 는 child 의 이벤트를 수신합니다.
    @Override
    public void onChildAdded(DataSnapshot dataSnapshot, String s)
    {
        ChatData chatData = dataSnapshot.getValue(ChatData.class);
        // chatData 를 가져오고
        adapter.add(chatData.getUserName() + ": " +
chatData.getMessage()); // adapter 에 추가합니다.
    }

    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String
s) { }

    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) { }

    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String s)
{ }

    @Override
    public void onCancelled(DatabaseError databaseError) { }
});
  
```

주 작업 진행 : 김민재

### 이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

너무 만만하게 보고 들어갔다가 방대한 양 때문에 깜짝 놀랐습니다

Firebird 는 굉장히 신통한 제품인 것 같아요. 이걸로 채팅도 만들 수 있고 데이터베이스도 할 수 있고.. 할 수 있는 건 많은데 안드로이드에 구현하는게 어려워요.

RecyclerView는 진짜 이해가 안되네요

Firebird 채팅에서 이번에는 사진을 올리고 내리는 코드를 넣지 못했습니다

다음 번 프로젝트에는 채팅창에서 이미지도 같이 처리할 수 있도록 하겠습니다

그리고 사용자의 지도상의 경로변화를 인터넷 DB에 올려서 언제든지 재현 해 볼 수 있는 기능을 넣고 싶습니다

지도교사 확인 및 의견	
<p>Firebird 가 인터넷에 공개된 자료도 적고 책의 제일 뒷부분에 나오는 숙련자를 위한 내용인데 잘 따라와 주었네요</p> <p>지금 구현한 기능을 다양하게 응용하면 될 것 같습니다.</p>	<p>(서명)</p> <p>허준성</p>

위치 추적용 센서 기능	
프로젝트 기간 및 목표	
<b>기간</b>	9월 1일 ~ 9월 20일
<b>목표</b>	영상 기반 위치 인식을 도와줄 보조 센서를 도입합니다
프로젝트 활동내용	
<p>이미지 방식의 네비게이션에서 영상을 찍을 때 만 위치를 알 수 있는데 스마트폰에 들어 있는 가속도 센서나 방향센서를 추가하여 사진을 안 찍는 순간에도 위치를 판단할 수 있으면 좋겠습니다</p> <p>(1) 방향 측정 - 폰을 들고 있는 방향을 나침반처럼 표시하는 기능입니다</p> <p>(2) 만보기 기능 - 스마트폰에서 만보기 기능은 기본적으로 지원되지 않으므로 몇가지 센서 정보를 합쳐서 만듭니다</p> <p>코드를 만들었는데 지도상의 사용자 위치와 연결을 시키지 못했습니다</p> <p>안드로이드의 Fragment 간의 데이터 전달이 어렵네요</p> <p>일단 구현한 코드는 다음과 같습니다</p> <pre> public class Sub5Activity extends AppCompatActivity implements SensorEventListener {     private CustomDrawableView mCustomDrawableView;     private SensorManager mSensorManager;     private Sensor accelerometer;     private Sensor magnetometer;     private float[] mGravity;     private float[] mGeomagnetic;      private Float azimuth; // View to draw a compass      protected void onCreate(Bundle savedInstanceState) {         super.onCreate(savedInstanceState);         mCustomDrawableView = new CustomDrawableView(this);         setContentView(mCustomDrawableView); // Register the sensor listeners         mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);         accelerometer = mSensorManager             .getDefaultSensor(Sensor.TYPE_ACCELEROMETER);         magnetometer = mSensorManager             .getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);          azimuth = 0.f; </pre>	

```

}

protected void onResume() {
    super.onResume();
    mSensorManager.registerListener(this, accelerometer,
        SensorManager.SENSOR_DELAY_UI);
    mSensorManager.registerListener(this, magnetometer,
        SensorManager.SENSOR_DELAY_UI);
}

protected void onPause() {
    super.onPause();
    mSensorManager.unregisterListener(this);
}

public void onAccuracyChanged(Sensor sensor, int accuracy) {
}

public void onSensorChanged(SensorEvent event) {
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER){
        mGravity = event.values;
    }
    if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD){
        mGeomagnetic = event.values;
    }
    if (mGravity != null && mGeomagnetic != null) {
        float R[] = new float[9];
        float I[] = new float[9];
        boolean success = SensorManager.getRotationMatrix(R, I,
            mGravity, mGeomagnetic);
        if (success) {
            float orientation[] = new float[3];
            SensorManager.getOrientation(R, orientation);
            azimuth = orientation[0]; // orientation contains: azimuth, pitch and roll
        }
    }
    mCustomDrawableView.invalidate();
}

public class CustomDrawableView extends View {
    Paint paint = new Paint();

    public CustomDrawableView(Context context) {
        super(context);
        paint.setColor(Color.RED);
        paint.setStyle(Paint.Style.STROKE);
        paint.setStrokeWidth(4);
        paint.setAntiAlias(true);
    };

    protected void onDraw(Canvas canvas) {

```

```

int width = getWidth();
int height = getHeight();
int centerx = width / 2;
int centery = height / 2;
float radius, dir;

if (centerx > centery) {
    radius = (float) (centery * 0.9);
} else {
    radius = (float) (centerx * 0.9);
}

// 방위각 구하기
dir = azimuth * 360 / (2 * 3.14159f);

// 나침판 그리기
canvas.drawCircle(centerx, centery, radius, paint);
canvas.drawLine(centerx, centery - radius, centerx, centery + radius, paint);
canvas.drawLine(centerx - radius, centery, centerx + radius, centery, paint);

// 방위 중앙에 표시
paint.setColor(Color.DKGRAY);
paint.setTextSize(300);
canvas.drawText(String.valueOf((int)dir), centerx-150, centery, paint);

// Canvas 를 회전 시킨다
if (azimuth != null){
    canvas.rotate(-dir, centerx, centery);
}

// 나침판 선 표시
paint.setColor(Color.BLUE);
paint.setTextSize(100);
canvas.drawLine(centerx, centery - radius, centerx, centery + radius, paint);
canvas.drawText("N", centerx - 30, centery - radius - 20, paint);
canvas.drawText("S", centerx - 30, centery + radius + 80, paint);
paint.setColor(Color.RED);
}
}
}

```

주 작업자 - 허채원

#### 이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

시간이 부족하여 위치인식에 붙이지를 못했습니다.

공모전과 별개로 위치 인식에 추가적인 데이터를 얻을 수 있는 센서 기능을 반드시 붙이겠습니다

지도교사 확인 및 의견	
<p>방향과 도보 스텝 기능을 구현하였는데 어떤 부분에 어떻게 붙여야 할지 결정하기 어려웠습니다</p> <p>좀더 아이들이 공리하여 완성도 있는 네비게이션이 되도록 업그레이드 하겠습니다</p>	<p>(서명)</p> <p>허준성</p>

## 지도 그리기

### 프로젝트 기간 및 목표

<b>기간</b>	7월 1일 ~ 8월 20일
<b>목표</b>	시장 지도를 화면상에 나타내고 애니메이션을 시킨다

### 프로젝트 활동내용

네비게이션에 시장 지도를 출력하는 모듈을 개발합니다

맵이 그려진 비트맵 파일을 안드로이드 폰에 뿌리고 건물에 해당하는 좌표를 내부적으로 기억하여 그 지점을 클릭하면 가게 정보를 보여주도록 만듭니다

사용자가 지나가는 길 정보를 보관하고 있다가 움직이는 사용자 애니메이션을 구현합니다

시행착오는 길 위에 사용자가 움직이게 만들었더니 다른 버튼들이 다 먹통이 되었어요

비트맵이 너무 큰가 싶어서 줄였는데 그대로였어요

며칠 고민하다가 지도교사님께 상의하고 Thread라는 해결책을 넘겨받았어요

현재 사용자 애니메이션은 잘 되고 있고요. 특정 지점 정보를 저장하는 것을 Firebase로 넘기는 것을 고려 중 입니다

```

public class GameViewBoy extends View {
    // Context, Thread
    private Context context;
    private GameThread mThread;

    // 배경, 화면 크기
    private Bitmap imgBack;
    private int w, h;

    // 소년
    private Boy myBoy;

    //-----
    // 생성자
    //-----
    public GameViewBoy(Context context, AttributeSet attrs) {
        super(context, attrs);

        // Context 저장
        this.context = context;
    }
  
```

```

//-----
// View 의 크기 구하기
//-----
@Override
protected void onSizeChanged(int w, int h, int oldw, int oldh) {
    super.onSizeChanged(w, h, oldw, oldh);

    // 화면의 폭과 높이
    this.w = w;
    this.h = h;

    // 배경 이미지
    imgBack = BitmapFactory.decodeResource(getResources(), R.drawable.map);
    imgBack = Bitmap.createScaledBitmap(imgBack, w, h, true);

    // 소년
    myBoy = new Boy(context, w, h);

    // 스레드 기동
    if (mThread == null) {
        mThread = new GameThread();
        mThread.start();
    }
}

//-----
// View 의 종료
//-----
@Override
protected void onDetachedFromWindow() {
    mThread.canRun = false;
    super.onDetachedFromWindow();
}

//-----
// 화면 그리기
//-----
@Override
protected void onDraw(Canvas canvas) {
    canvas.drawBitmap(imgBack, 0, 0, null);

    // 그림자
    //canvas.save();
    //canvas.scale(myBoy.shadowScale, myBoy.shadowScale, myBoy.x, myBoy.ground);
    //canvas.drawBitmap(myBoy.shadow, myBoy.x - myBoy.sw, myBoy.ground - myBoy.sh,
null);
    //canvas.restore();

    // 소년
    //canvas.scale(0.1f, 0.1f);
    //canvas.drawBitmap(myBoy.boy, myBoy.x - myBoy.w, myBoy.y - myBoy.h, null);

```

```

    canvas.drawBitmap(myBoy.boy, myBoy.x, myBoy.y, null);
  }

  //-----
  // 소년 이동
  //-----
  private void moveBoy() {
    myBoy.update();
  }

  //-----
  // Touch Event
  //-----
  @Override
  public boolean onTouchEvent(MotionEvent event) {
    if (event.getAction() == MotionEvent.ACTION_DOWN)
    {
      // myBoy.jump();
      String ss = "pos[" + event.getX() + " , " + event.getY();
      Toast.makeText(getContext(), ss, Toast.LENGTH_SHORT).show();

      myBoy.x = event.getX();
      myBoy.y = event.getY();
    }

    return true;
  }

  //-----
  // Thread
  //-----
  class GameThread extends Thread {
    public boolean canRun = true;

    @Override
    public void run() {
      while (canRun) {
        try {
          Time.update(); // deltaTime 계산

          moveBoy();
          postInvalidate(); // 화면 그리기
          sleep(10);
        } catch (Exception e) {
          //
        }
      }
    }
  } // Thread
} // GameView

```

다음은 지도를 불러와서 표시한 화면입니다

이 길위에서 사용자가 움직입니다



### 이번 프로젝트에 대한 반성 / 다음 프로젝트 계획

사용자가 자신만의 지도를 편하게 만들 수 있는 도구를 만들어주지 못했어요

또한 공모전과는 상관없이 지도 이미지와 벡터정보를 쉽게 넣을 수 있는 기능을 추가하려 합니다  
지도는 확대 기능을 못 넣었어요. Canvas 그림을 그리는데 확대 기능을 넣으면 건물 좌표계산이  
어렵고 시간이 부족하여 보고서 제출후에 처리하려 합니다

주작업자 - 허채영	
<b>지도교사 확인 및 의견</b>	
<p>지도 상에 많은 기능을 넣을 수 있는데 그러지 못했습니다. 기본 animation과 축소 확대 공부에 시간을 많이 사용했네요 본인이 사용자 입력 기능을 넣겠다고 하니 도와주겠습니다</p>	<p>(서명) 허준성</p>
<b>프로젝트2 (프로젝트명)</b>	