

# Native App 프로젝트 기술서 - 1

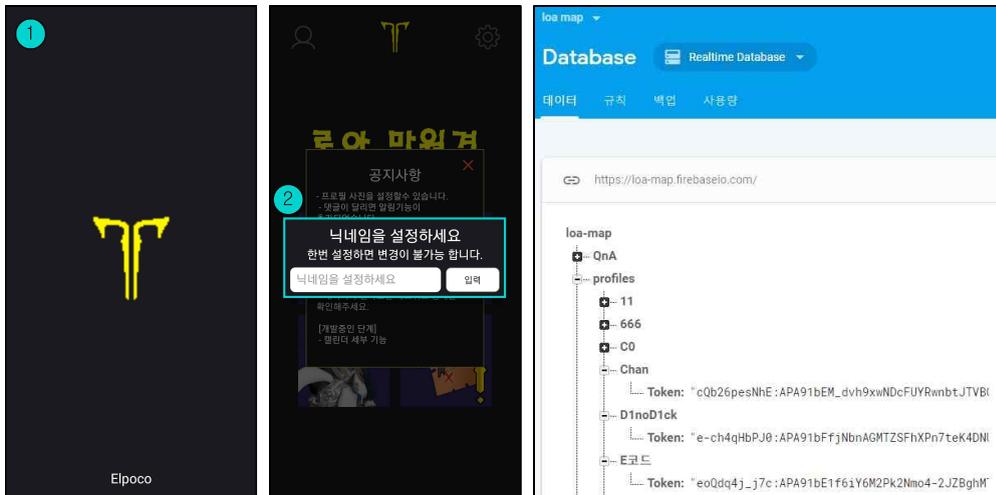
## 1) 프로젝트 개요

프로젝트 명	로아 망원경 (게임 정보 공유 앱)	
개발기간	2019.02.18.~2019.03.15. (약 3주)	
개발인원	1인 개발	
담당역할	기획, 관리, 개발, 디자인, 서버	
개발환경	OS	Windows 7
	Development Tool	Android Studio Adobe Photo Shop Adobe Illustrator
	Language	Java, XML, PHP
	DB	MySQL, Firebase실시간데이터베이스
마켓주소	<a href="https://play.google.com/store/apps/details?id=com.elpoco.p_mapfinder">https://play.google.com/store/apps/details?id=com.elpoco.p_mapfinder</a>	

## 2) 프로젝트 소개

프로젝트 설명
<p>로아 망원경은 게임정보를 공유하는 애플리케이션입니다. 게임 내 상점, 지도 힌트, 각종 콘텐츠를 확인 할 수 있으며, 지도 게시판에 사용자들이 게시물을 올려 다른 사용자들에게 공유 할 수 있습니다. 게임 정보를 저장하기 위해 res폴더의 values-arrays.xml로 각종 데이터를 저장하였고, 실시간으로 바뀌는 콘텐츠는 홈페이지의 HTML을 파싱하였습니다.</p> <p>게시물을 저장하기 위해 MySQL을 이용하였고, 사용자의 프로필 이미지등을 저장하기 위해 Firebase의 실시간 데이터베이스를 이용하였습니다.</p> <p>댓글이 달리게 되면 바로 알수 있게 Firebase CloudMessging를 사용해 알림 기능을 사용하였으며, 각종 설정과 사용자 정보를 기기에서도 저장할 수 있게 Shared Preference를 사용했습니다.</p>

## 주요화면 및 설명



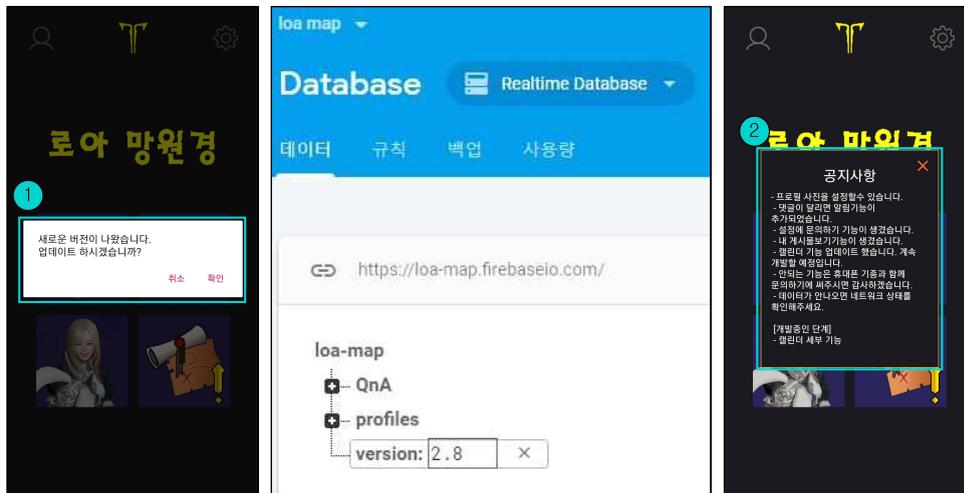
### 1. Intro

- 애플리케이션 실행 시 나오는 첫 화면입니다.
- Intro 화면에서 Shared Preferences에 저장되어있는 정보를 불러옵니다.

### 2. 닉네임 설정 (Firebase 실시간 데이터베이스)

- 첫 실행 시 닉네임을 설정하는 다이얼로그가 나옵니다.
- 닉네임을 설정하면 토큰값과 함께 Firebase 실시간 데이터베이스와 SharedPreference에 저장됩니다.

## 주요화면 및 설명



### 1. 최신버전 확인하기

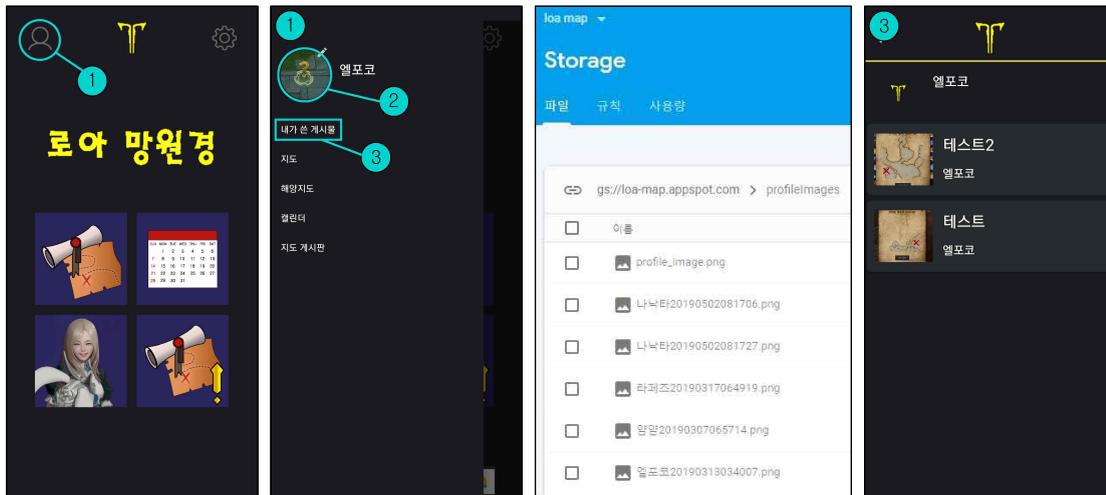
- 마켓에 올라온 버전과 설치되어 있는 버전이 다르면 업데이트 다이얼로그를 띄웁니다.
- Firebase 실시간 데이터베이스에 버전정보를 저장, 애플리케이션 실행 할 때 마다 버전체크를 합니다.

### 2. 공지사항

- 업데이트 내역을 보여줍니다. Layout으로 만들었으며 첫 실행 시에는 visible상태입니다.
- 2중 구조의 Layout으로 설계를 하여 공지사항 밖을 클릭하면 이벤트가 발동 안 되도록 하였습니다.
- 애플리케이션의 버전을 구해 Shared Preferences에 저장되어 있는 버전보다 최신일 경우 Visible상태로 됩니다.

- 오른쪽 위에 있는 X버튼을 클릭할 경우 Shared Preference에 버전을 최신화 합니다.

### 주요화면 및 설명



#### 1. 내 정보 (Navigation View)

- 메인 화면 중앙에 있는 버튼을 Navigation View에서도 들어갈 수 있게 하였습니다.

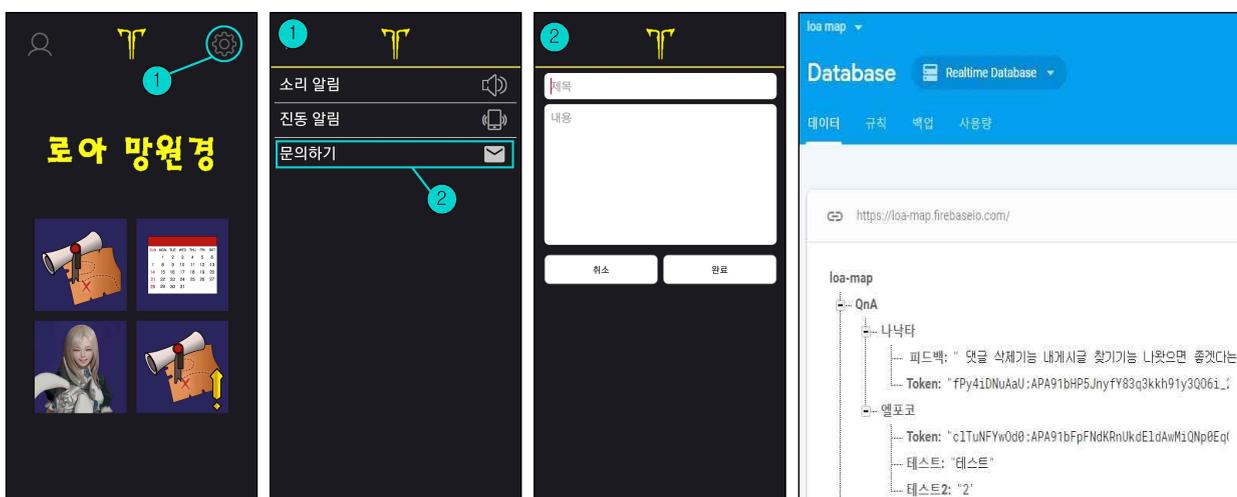
#### 2. 프로필 이미지 (Firebase Storage)

- 설정을 안 하게 될 경우 기본 이미지로 설정 됩니다.
- 클릭 할 경우 Gallery를 Intent합니다. 사진을 선택하면 프로필 이미지를 변경할 수 있습니다.
- 사진을 선택하면 이미지가 변경되고, 이미지가 Firebase Storage에 저장이 됩니다.

#### 3. 내가 쓴 게시물 (MySQL)

- 메인화면에 없는 버튼인 내가 쓴 게시물 화면으로 이동할 수 있게 맨 위에 배치하였습니다.
- 게시물마다 저장되어있는 토큰 값을 비교해 사용자와 같은 값이라면 list형식으로 불러옵니다.
- 클릭 하게 되면 게시물 화면으로 이동합니다.

### 주요화면 및 설명



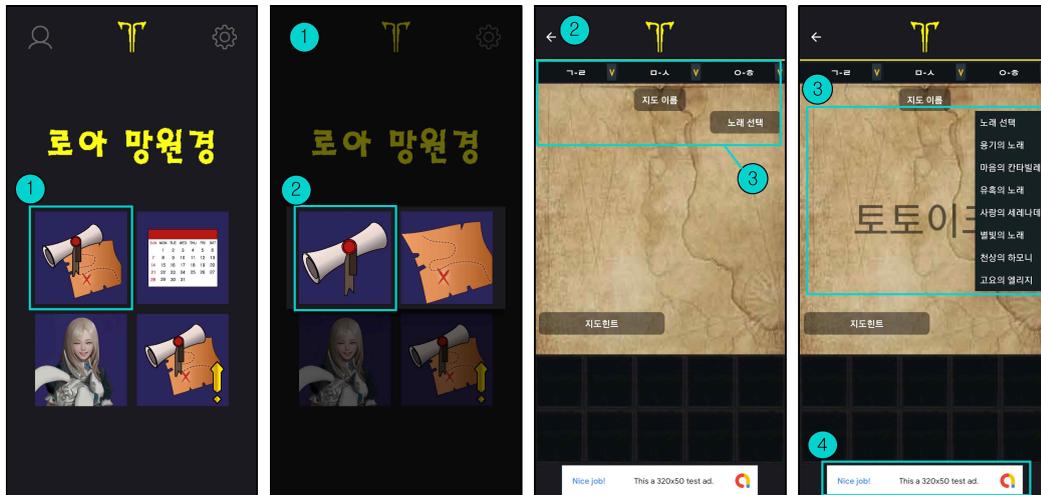
#### 1. 설정

- 알림 설정을 할 수 있으며 사용자들의 불편함이나 요청사항을 문의하기를 통해 확인할 수 있습니다.

#### 2. 문의하기 (Firebase 실시간 데이터베이스)

- 글을 작성하게 되면 Firebase 실시간 데이터베이스 올라가며 여러 번 문의를 할 경우 밑으로 node가 더 생기게 됩니다.

## 주요화면 및 설명



### 1. 지도힌트, 해양지도

- 지도 힌트와 해양지도를 확인할 수 있는 버튼입니다.
- 클릭을 하게 되면 dialog가 나오게 됩니다.

### 2. 지도 힌트화면

- LinearLayout으로 구성을 하여 구현하였습니다.

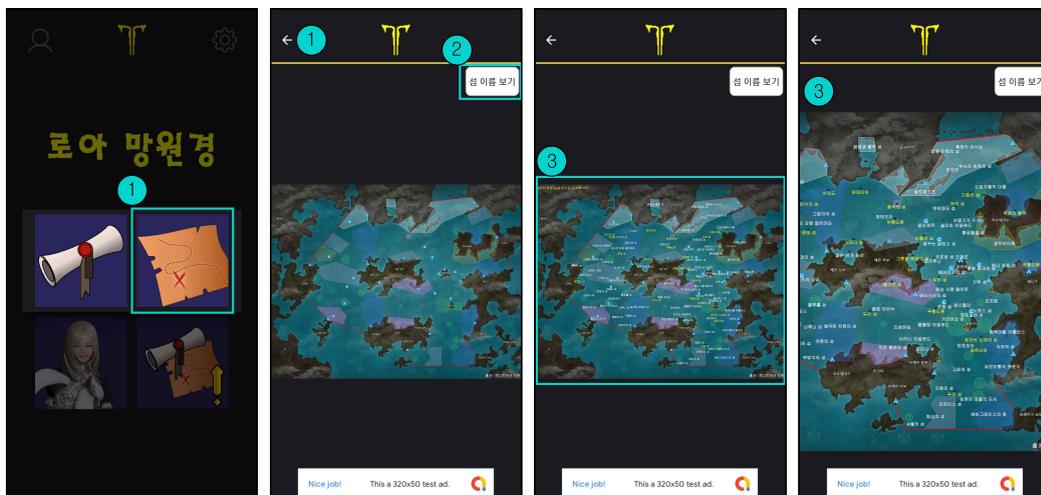
### 3. 힌트/답

- 힌트를 보여주는 목록은 spinner로 구성되어 있습니다.
- 힌트를 클릭 하면 힌트에 맞는 지역이름이 나오게 됩니다. 힌트와 답은 배열로 구성되어 있습니다.

### 4. AdMob

- 구글의 AdMob플러그인을 통해 하단부에 Banner광고를 삽입했습니다.

## 주요화면 및 설명



### 1. 해양지도 화면

- 해양지도를 보여주며 해양컨텐츠를 확인할 수 있습니다.
- RelativeLayout으로 구성하여 지도를 중앙에 배치하였습니다.

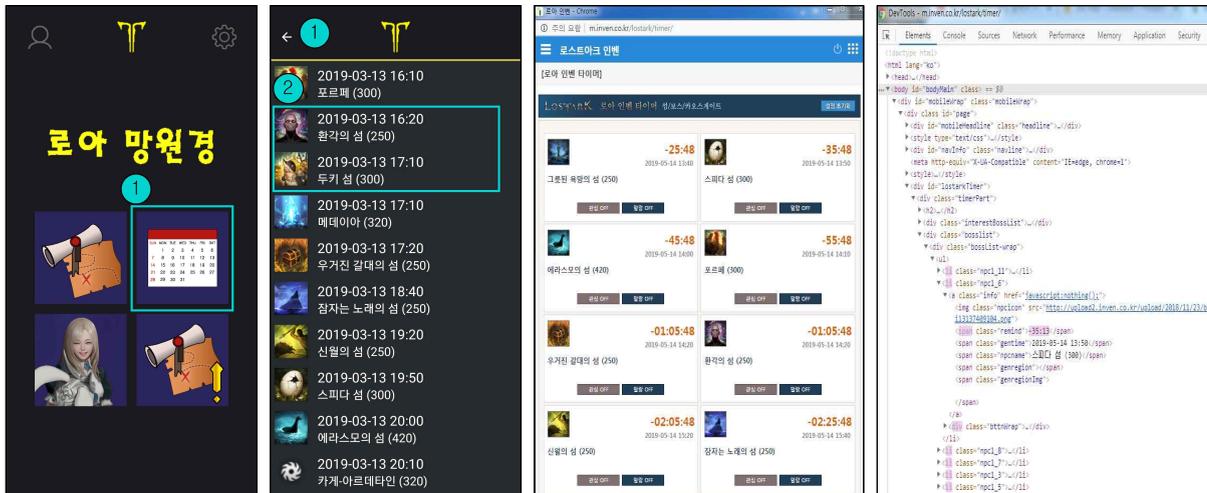
### 2. 섬 이름 보기 버튼

- 각 섬들의 이름을 확인할 수 있습니다. 사진 두 개를 교대로 보여줍니다.

### 3. 해양지도 확대/축소

- 외부 라이브러리인 photoView를 사용하여 보기 편하게 확대/축소를 할 수 있습니다.

#### 주요화면 및 설명



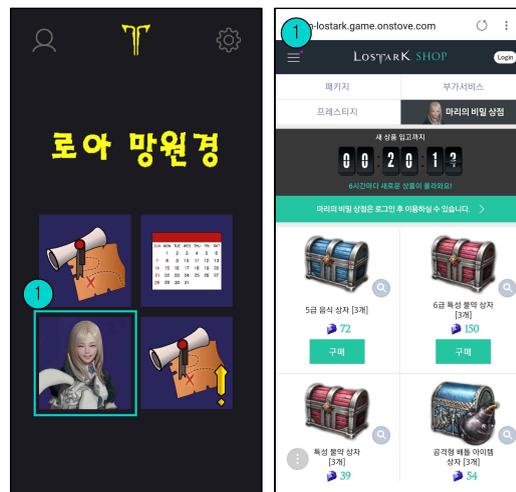
#### 1. 캘린더 (HTML 파싱)

- 로스트 아크 게임에 시간마다 즐길 수 있는 컨텐츠/섬을 확인할 수 있습니다.
- RecyclerView를 사용했으며 시간마다 화면이 갱신되어야 하기에 Thread를 사용하여 15초마다 갱신되도록 구현하였습니다.

#### 2. 캘린더 자료

- 로스트 아크 인벤 사이트에 있는 타이머를 HTML파싱을 하였습니다.
- 각 요소의 속성을 식별해 원하는 값을 가져옵니다.

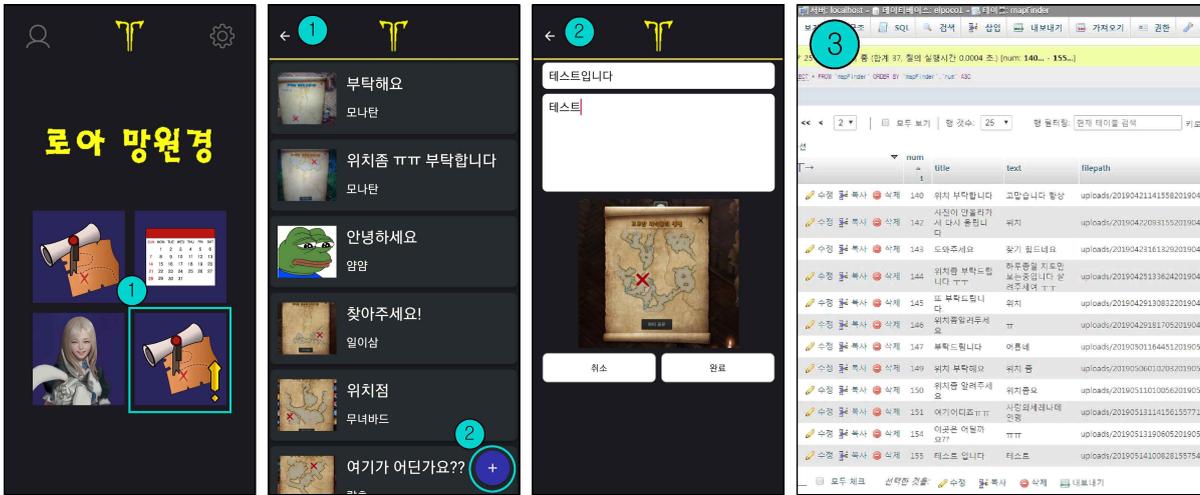
#### 주요화면 및 설명



#### 1. 마리상점 (게임 내 상점)

- 로스트아크 홈페이지에 상점 링크를 사용해 intent 하여 화면을 보여줍니다.

## 주요화면 및 설명



### 1. 지도 게시판 (MySQL)

- MySQL에 저장되어 있는 게시물의 데이터를 불러와 리스트로 보여줍니다.
- 10개의 글을 보여주며 맨 아래까지 스크롤 할 경우 5개씩 글을 더 보여줍니다.

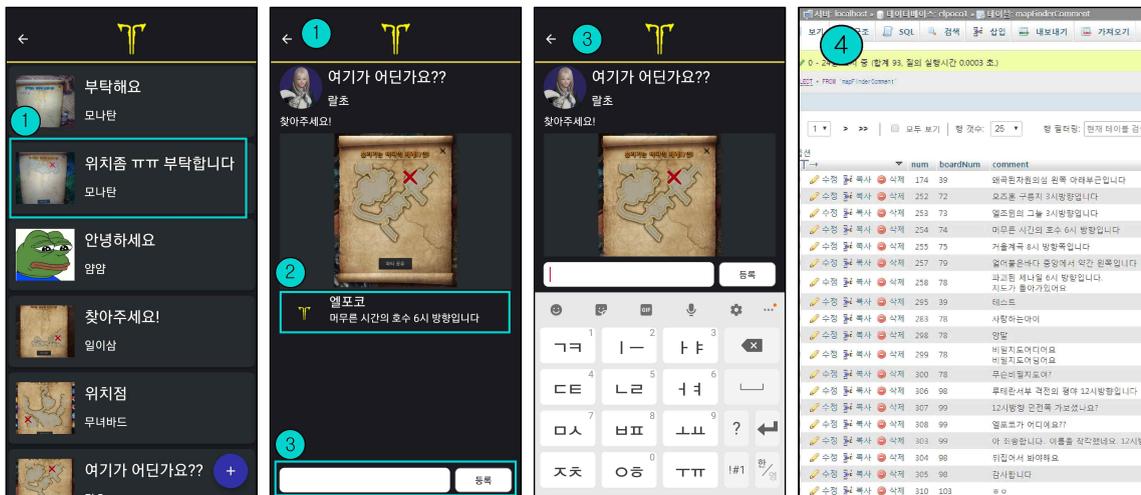
### 2. 게시물 작성

- 게시물 작성을 하게 되면 서버에 있는 PHP언어를 통해 서버에 게시물이 올라가게 됩니다.
- 사진은 dothome서버의 DB에 저장됩니다.

### 3. MySQL

- 게시물이 저장되어있는 서버입니다.
- 게시물번호, 제목, 내용, 닉네임, 프로필사진, 업로드 사진이 저장됩니다.

## 주요화면 및 설명



### 1. 게시물

- 클릭을 하게 되면 item의 데이터들을 Intent의 putExtra메소드를 사용해 데이터를 넘깁니다.

### 2. 댓글

- 서버에 저장된 댓글들을 불러와 리스트로 보여줍니다.
- 댓글 테이블의 게시물번호를 식별하여 게시물의 번호가 같은 댓글만 불러옵니다.

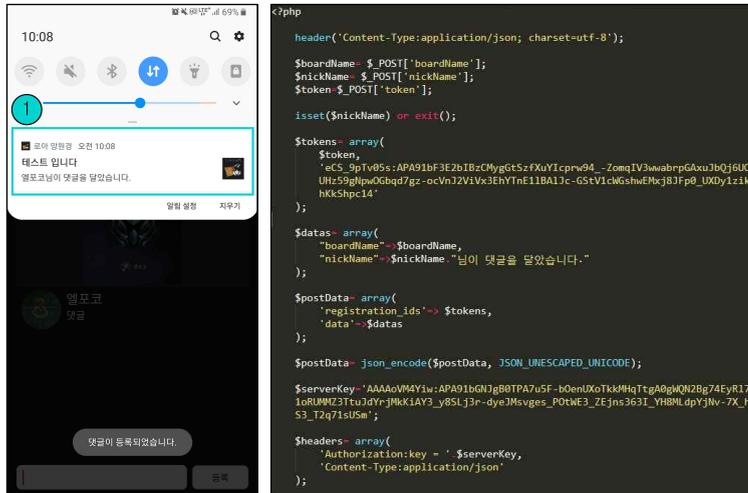
### 3. 댓글쓰기

- 댓글을 작성하면 서버에 게시물의 번호와 댓글, 닉네임, 프로필 사진을 서버에 저장합니다.

#### 4. MySQL

- 댓글 저장되어 있는 서버입니다. 게시물 번호, 댓글, 닉네임, 프로필사진의 경로를 저장합니다.

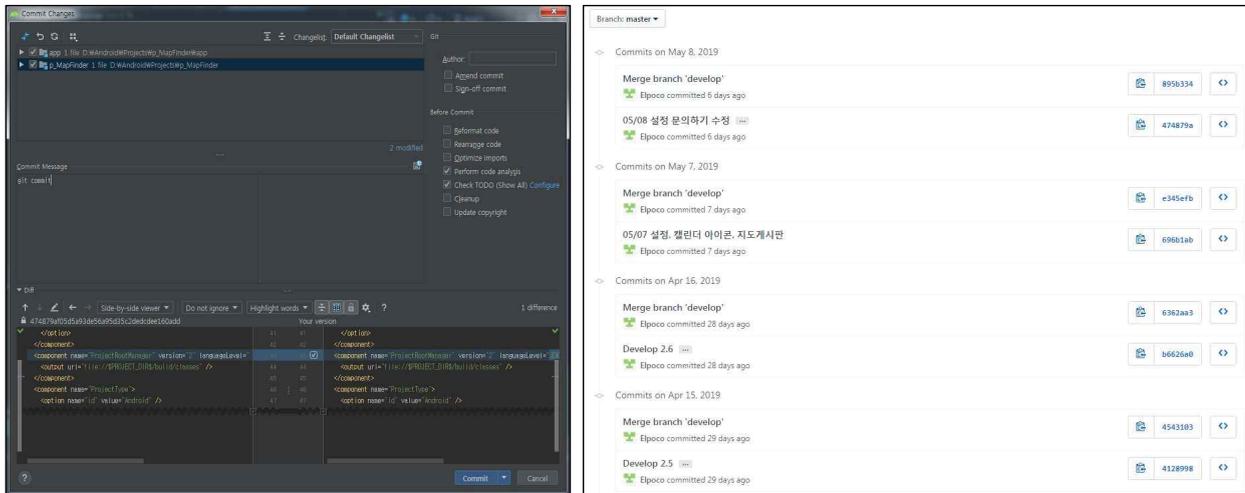
### 주요화면 및 설명



#### 1. 댓글 알림 (Firebase Cloud Messaging)

- 게시물에 댓글이 달리게 되면 그 게시물의 토큰값을 얻어와 서버에 저장되어있는 PHP문서를 실행시켜 게시물을 쓴 사용자에게 알림을 보냅니다.

### 주요화면 및 설명



#### 1. GitHub

- GitHub Commit하기
- GitHub Merge하기

## Native App 프로젝트 기술서 - 2

### 1) 프로젝트 개요

프로젝트 명	SpinMon (게임 앱)	
개발기간	2019.01.09.~2019.01.17. (약 1주)	
개발인원	1인 개발	
담당역할	기획, 관리, 개발, 디자인	
개발환경	OS	Windows 7
	Development Tool	Android Studio
	Language	Java, XML
마켓주소	<a href="https://play.google.com/store/apps/details?id=com.elpoco.spinmon">https://play.google.com/store/apps/details?id=com.elpoco.spinmon</a>	

### 2) 프로젝트 소개

#### 프로젝트 설명

SpinMon은 수업시간동안 배웠던 기술들을 사용해 제작한 게임 애플리케이션입니다. 게임에 사용된 이미지는 DragonFlight의 이미지를 사용하였으며, 학습용으로 개발하였습니다. 원활한 게임 플레이를 위하여 Surface View를 상속받아 화면을 구현하였으며, 디바이스 성능차이에 따른 게임 진행속도를 조절하기 위하여 FPS기능을 사용하였습니다. 게임 스코어와 각종 설정값을 저장하기 위해 SharedPreferences를 사용하였습니다.

#### 주요화면 및 설명



#### 1. MainActivity

- MediaPlayer 클래스를 사용하여 게임의 배경음악을 반복 재생 설정하였습니다.
- GameActivity로 이동하기전 사용자가 저장한 게임설정값을 Shared Preference클래스를 통하여 Static 변수들이 저장된 G클래스에 저장하였습니다.

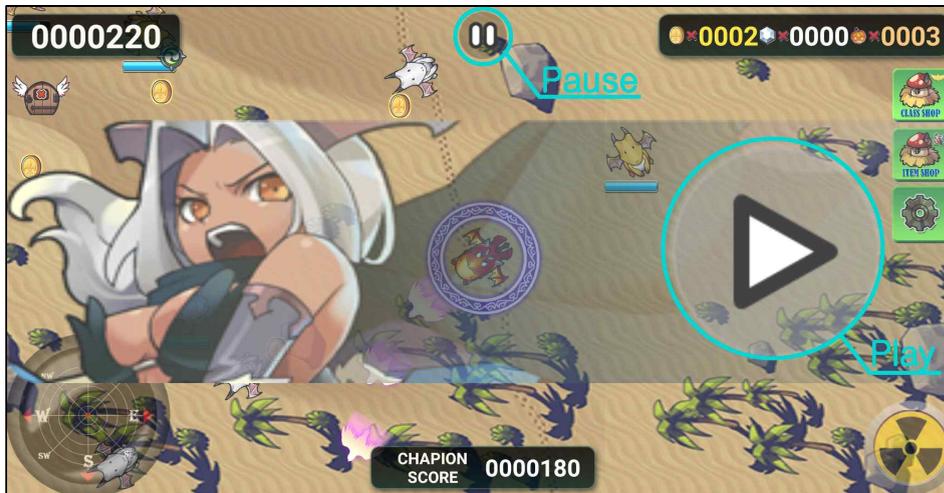
## 주요화면 및 설명



### 1. GameActivity

- 게임 진행을 보여주기 위해 적절한 RelativeLayout을 사용하여 요소들을 배치하였습니다.
- 게임의 진행을 보여주는 점수, 코인들은 ImageView, TextView를 사용하였습니다.
- 게임의 배경이나 몬스터, 효과처럼 움직임이 많은 요소들은 SurfaceView를 상속받아 onDraw 메소드를 오버라이드 하여 구현 하였습니다.
- 미사일과 몬스터의 중심좌표를 구하여 각 중심좌표를 Math.pow로 곱치는 여부를 판별하여 충돌 여부에 따른 효과를 구현하였습니다.
- 몬스터를 잡으면 아이템, 코인이 확률적으로 나오게 되며, 미사일과 몬스터, 플레이어의 중심좌표를 구하여 Math.pow를 사용해 곱치는 여부를 판별하여 충돌 여부에 따라 Switch-case문으로 피격효과와 아이템효과를 구현하였습니다.
- 플레이어와 몬스터가 닿게 되면 몬스터는 사라지고, 플레이어의 HP가 줄어들게 됩니다. 이때, 몬스터와 닿았다는 느낌을 주기 위하여 진동이 울리게 설정 하였습니다.

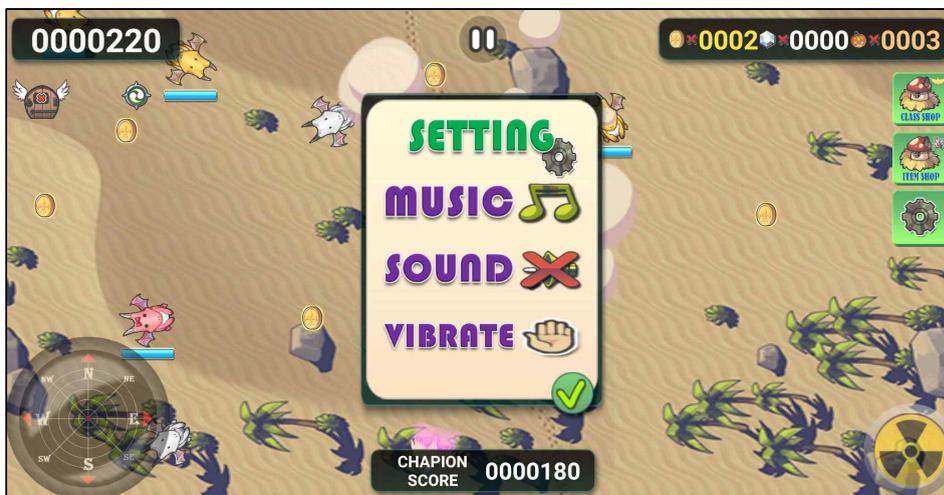
## 주요화면 및 설명



### 1. GameActivity (Pause)

- Pause버튼을 클릭하면 Thread의 wait메소드를 실행시켜 게임을 멈춥니다.
- Play버튼을 클릭하면 Thread가 다시 실행되어 게임이 진행됩니다.

## 주요화면 및 설명



### 1. GameActivity (Setting)

- 게임의 옵션을 설정할 수 있는 ToggleButton 클래스들에 리스너를 구현하여 Switch-case문을 사용하여 기능을 구현하였습니다.
- 매번 설정할 필요 없도록 Shared Preference에 저장이 되도록 하였습니다.

## 주요화면 및 설명



### 1. GameOverActivity

- 플레이어의 HP가 모두 줄어들면 GameOverActivity로 전환됩니다.
- 이미지를 클릭하게 되면 Gallery를 Intent하여 사진을 선택할 수 있습니다.
- 게임을 플레이 하면서 얻었던 코인들을 Shared Preference에 저장합니다.

# Native App 프로젝트 기술서 - 3

## 1) 프로젝트 개요

프로젝트 명	경기도 교통사고 (Open API XML파싱)	
개발기간	2018.12.24. (1일)	
개발인원	1인 개발	
담당역할	기획, 관리, 개발, 디자인	
개발환경	OS	Windows 7
	Development Tool	Android Studio Adobe Illustrator
	Language	Java, XML
마켓주소	<a href="https://play.google.com/store/apps/details?id=com.elpoco.xmlparsingex">https://play.google.com/store/apps/details?id=com.elpoco.xmlparsingex</a>	

## 2) 프로젝트 소개

### 프로젝트 설명

경기도 교통사고는 Open API 수업을 한 후 학습을 위해 만든 애플리케이션입니다. 정보들을 보기 편하게 리스트로 구현을 하였고, 스크롤을 마지막까지 내릴 때마다 정보를 추가로 불러옵니다.

### 주요화면 및 설명

### 1. MainActivity (XML파싱)

- LOAD버튼을 누르면 데이터 10개를 불러옵니다.
- 데이터의 각 요소를 switch-case 문으로 원하는 값을 가져와서 ArrayList에 저장합니다.
- 리스트의 개수와 스크롤의 위치를 계산해 데이터를 다 불러온 경우 10개씩 데이터를 불러오게 됩니다.

# Java 프로젝트 기술서

## 1) 프로젝트 개요

프로젝트 명	Catch Mind (게임)	
개발기간	2018.11.19.~2018.11.30. (약 1주)	
개발인원	1인개발	
담당역할	기획, 관리, 개발, 디자인	
개발환경	OS	Windows 7
	Development Tool	Eclipse Adobe Illustrator 그림판
	Language	Java

## 2) 프로젝트 소개

### 프로젝트 설명

넷마블의 캐치마인드를 보고 아이디어를 얻어 만들어본 JAVA 프로젝트입니다. 플레이어가 출제자가 되어 제시어를 보고 마우스로 그림을 그리고, 그 그림을 다른 플레이어들이 정답을 채팅창으로 정답을 맞히는 게임입니다. 플레이어가 정답을 맞으면 그 플레이어가 출제자가 되고 제시어도 바뀌게 됩니다. Swing을 사용하여 GUI를 구현하였고, 각종 데이터들은 서버로 특정 문자열과 함께 보내지며 받은 데이터들을 클라이언트로 보내게 됩니다. 데이터를 받은 클라이언트는 문자열을 판별해 그림그리기, 플레이어 체크, 정답확인, 채팅 기능을 합니다.

### 주요화면 및 설명



#### 1. MainClient

- 서버와 클라이언트를 실행 시킬 수 있습니다.

#### 2. ServerFrame

- 클라이언트의 데이터를 받아서 다른 클라이언트에게 보낼 수 있는 서버입니다.

#### 3. ClientFrame

- 아이디를 입력해 클라이언트를 실행 시킬 수 있습니다.

## 주요화면 및 설명

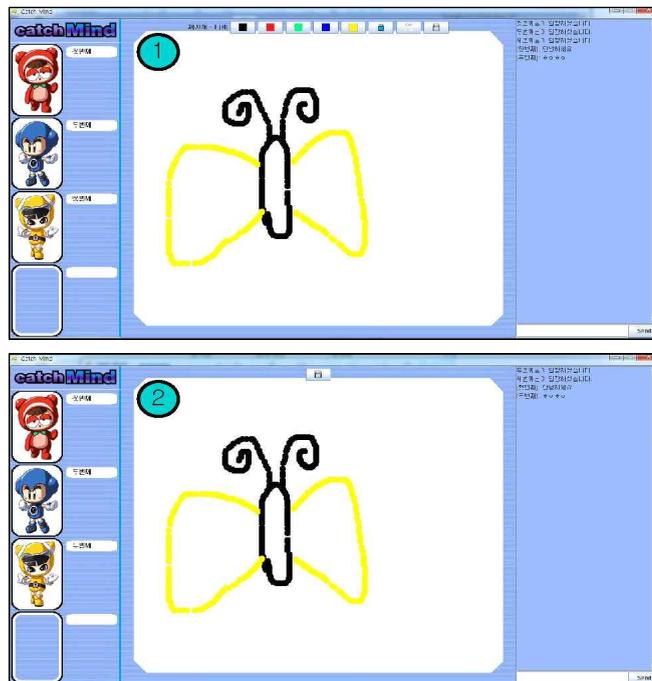


```
void setID(int playernum, String[] names) {  
    switch(playernum) {  
        case 4:  
            Player4.nickname.setText(names[3]);  
        case 3:  
            Player3.nickname.setText(names[2]);  
        case 2:  
            Player2.nickname.setText(names[1]);  
        case 1:  
            Player1.nickname.setText(names[0]);  
            break;  
        default:  
            break;  
    }  
}
```

### 1. 플레이어

- 플레이어의 입장/퇴장 시 마다 해당되는 자리를 수정합니다.
- switch-case문으로 구현 하였고 입장 시 닉네임과 소켓 배열의 위치로 해당되는 자리를 보여줍니다.

## 주요화면 및 설명



```

dataOutputStream.writeUTF("$"+ID+"님이 입장하셨습니다.\n");
dataOutputStream.flush();
while(true) {
    inmsg=dataInputStream.readUTF();
    if(inmsg.equals("!dispose!") disposeClient();
    type=inmsg.substring(0, 4);
    if(type.equals("draw")) {
        value=inmsg.substring(4);
        addHost(value);
    }
    else {
        if(inmsg.contains("#player")) { // 플레이어가 입장했을때
            playerNum = Integer.parseInt(inmsg.substring(inmsg.indexOf("#")+1));
            setImage(playerNum);
            continue;
        }
        if(inmsg.contains("****")) { // 플레이어 위치
            myNumber=Integer.parseInt(inmsg.substring(inmsg.lastIndexOf("**")+1)-1);
            inmsg=inmsg.replace("****", "");
            continue;
        }
        if(inmsg.contains("-exit")) { // 플레이어가 퇴장했을때
            inmsg=inmsg.replace("-exit", "");
            playerNum=Integer.parseInt(inmsg);
            exitID(playerNum, names);
            continue;
        }
        if(inmsg.contains("$")) inmsg=inmsg.replace("$", ""); // 플레이어 아이디 확인
        if(inmsg.contains("&&")) { // 플레이어 아이디 목록 받아오기
            inmsg=inmsg.substring(3, inmsg.length()-1);
            names=inmsg.split(", ");
            setID(names.length, names);
            continue;
        }
        if(inmsg.contains("clear_user")) {drawPanel.clear(); continue;}
        if(inmsg.contains("-question")) { // 제시어 변경하기
            inmsg=inmsg.replace("-question", "");
            question=inmsg.substring(inmsg.indexOf("#"), inmsg.lastIndexOf("#")+1);
            inmsg=inmsg.replace(question, "");
            question=question.replaceAll("#", "");
        }
        if(inmsg.contains("-turn")) {turn=true; drawPanel.btnOnOff(); inmsg=inmsg.repl
        if(inmsg.contains("[+ID+]")) continue; // 자기가 보낸 메시지는 받지 않기
        if(inmsg.contains("-answer")) {drawPanel.clear(); turn=false; drawPanel.btnOnOff

```

**1. 출제자**

- 출제자만 그림을 그릴 수 있습니다.
- 그림을 그리게 되면 MouseMotionListener를 상속받은 JPanel에 마우스 위치를 좌표로 구해와 구해온 좌표에 점을 찍습니다.
- 서버로 좌표 값이 이동하며, 서버에서 다른 클라이언트에 좌표 값을 보내게 됩니다.

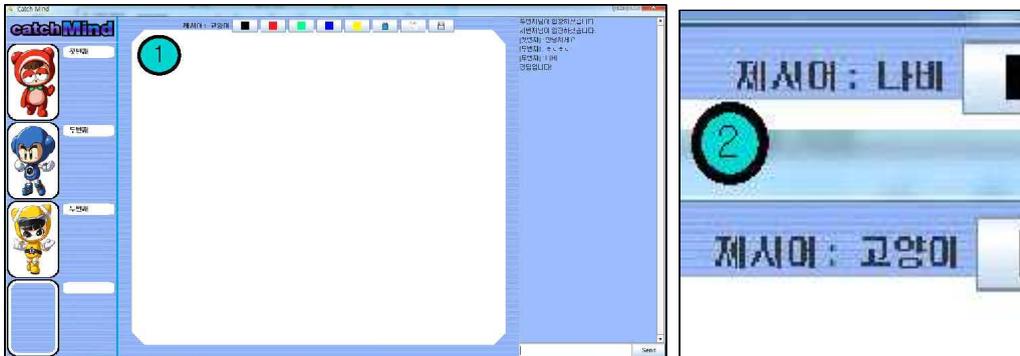
**2. 플레이어**

- 서버에서 온 좌표 값을 배열에 저장하여 그림을 그리게 됩니다. 플레이어들은 그림을 그릴 수 없습니다.

**3. Code**

- 문자열과 결합된 데이터들을 판별하고, String의 substring메소드를 이용하여 작동합니다.

**주요화면 및 설명**



**1. 정답**

- 정답을 맞힐 경우 정답을 맞힌 플레이어가 출제자로 변경 됩니다.
- 서버로 정답을 맞힌 플레이어의 번호를 보내고, 다시 모든 클라이언트로 보냅니다.
- 서버로 온 문자열을 판별하여 숨겨진 버튼을 보이게 하고 그림을 그릴 수 있게 합니다.

**2. 제시어**

- 정답을 맞히게 되면 제시어가 저장되어있는 배열에서 랜덤으로 값을 가져와 바꾸게 됩니다.