



컴퓨터정보공학과

졸업작품 포트폴리오



2021-11-27

20185053 김기태 PA

Contents

- 졸업작품 제작하면서 사용한 기능들
- 졸업작품을 진행하면서 공부한 내용
- 졸업작품 완료 후 느낀 점

이번 졸업작품을 하면서 5 조가 쓴 톨은 90 퍼센트가 안드로이드 스튜디오이다.

졸업작품을 제작하면서 사용한 기능

안드로이드 스튜디오에서 사용한 기능을 정리하면 ,로그인 기능, 화면 넘기기, 서버에 데이터 주고받기, 카카오,구글 등과 안드로이드 api 연동하기, 라운드가 넘어갈 때 마다

이미지 버튼 이미지 변경하기, 랜덤 으로 이미지와 텍스트 내보내기 등이 있다.

원래 하려던 오늘의 앱이 기술적 오류로 불발되었고 새로 시작하게 된 오늘 뭐먹지는 준비 할 기간이 많이 줄었었기 때문에 정말 어려운 코딩 없이 만들게 되어서 이렇다할 기능이 별로 없다.

먼저 화면단을 이동하기 위해서 intent 를 통해 로그인부터 메인, 메뉴선택화면 등을 이동하게된다.

```
public void onClick(View v) {
    Intent intent = new Intent( packageContext: MainActivity.this, Menus.class);
    startActivity(intent);
}

btn_logout = findViewById(R.id.btn_logout);
logout.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // 로그아웃
        mFirebaseAuth.signOut();
        Intent intent = new Intent( packageContext: MainActivity.this, LoginActivity.class);
        startActivity(intent);
        finish();
    }
})
```

```

btn_kinds = (Button) findViewById(R.id.choose1);
btn_sit = (Button) findViewById(R.id.choose2);
btn_wc = (Button) findViewById(R.id.choose3);

btn_kinds.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent intent = new Intent( packageContext: Menus.this, Kinds.class);
        startActivity(intent);
    }
});

btn_sit.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent intent = new Intent( packageContext: Menus.this, Situation.class);
        startActivity(intent);
    }
});

btn_wc.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent intent = new Intent( packageContext: Menus.this, SelectActivity.class);
        startActivity(intent);
    }
});

```

코드를 보게 되면 버튼에 Intent 를 주어 버튼을 눌렀을 때 setOnClickListener 함수 안에 파라미터값으로 들어가있는 class(액티비티)로 이동하게 된다. 매우간단한 기능이다.

안드로이드 스튜디오에서는 api 를 사용하거나 서버에 연동, 라이브러리 기능등 외부에 것들을 끌어 와서 사용할 수가 있는데 이것들은 모두 gradle(app)의 dependencies 에서 종속으로 implementation 해주어 sync 를 해야 사용이 가능하다

뿐만 아니라 SDK(컴파일)의 버전, 빌드 버전 등도 설정을 해주어야 한다. Api 를 사용할 때는 JSON 형식으로 받아오게 되는데 Parsing 을 꼭 해주어야 한다. 먼저 JSON 이란 **J**ava **S**cript **O**bject **N**otation 의 약자로 "네트워크를 통해 데이터를 주고받는데 사용되는 경량의 데이터 형식"이다.

JSON 은 **name : value** 형식으로 구성되어있으며 name 에는 String 값, value 에는 기본 자료형 혹은 배열,객체 등이 될 수 있다. JSON 파싱은 이 데이터들을 이용하기 위해 데이터만 뽑아서 배열, 딕셔너리 형태로 재구성하여 저장하는 것을 말한다.

Json 파일을 읽어와파일내용을 String 변수에담아 return 하는함수이다

```
private String getJsonString()
{
    String json = "";

    try {
        InputStream is = getAssets().open("Movies.json");
        int fileSize = is.available();

        byte[] buffer = new byte[fileSize];
        is.read(buffer);
        is.close();

        json = new String(buffer, "UTF-8");
    }
    catch (IOException ex)
    {
        ex.printStackTrace();
    }

    return json;
}
```

```

public class Movie{
    private String title;
    private String Category;
    private String grade;

    public String getTitle() {
        return title;
    }

    public String getCategory() {
        return Category;
    }

    public String getGrade() {
        return grade;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public void setCategory(String category) {
        Category = category;
    }

    public void setGrade(String grade) {
        this.grade = grade;
    }
}

```

Getter/setter 함수를 구현하여 각 멤버변수에 접근할 수 있다

```

private void jsonParsing(String json)
{
    try{
        JSONObject jsonObject = new JSONObject(json);

        JSONArray movieArray = jsonObject.getJSONArray("Movies");

        for(int i=0; i<movieArray.length(); i++)
        {
            JSONObject movieObject = movieArray.getJSONObject(i);

            Movie movie = new Movie();

            movie.setTitle(movieObject.getString("title"));
            movie.setGrade(movieObject.getString("grade"));
            movie.setCategory(movieObject.getString("category"));

            movieList.add(movie);
        }
    }catch (JSONException e) {
        e.printStackTrace();
    }
}

```

JSONObject를 생성한다 인자로 JSON의 String형 변수를 넘겨준다. 그후 받아오려는 데이터의 value에 해당하는 배열에 대한 정보를 getJSONArray()함수를 호출해 JSONArray 객체를 생성하고 JSONArray를 순회하며 각 데이터 정보를 가져온다 getObject함수를 통해 가져올 데이터 정보에 해당하는 객체를 Object객체가 참조하도록 한다. key값을 가져오기 위해서는 getString() 함수를 호출하여 인자 정보로 가져오고자 하는 값에 해당하는 key값을 넘겨주게 된다;

그 다음은 랜덤으로 데이터들 중 하나의 값을 출력해주는 코드이다.

```
imageView = findViewById(R.id.imageView);
btn_click = (Button) findViewById(R.id.click);
txtResult = findViewById(R.id.food_name);
btn_click.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        int count = random.nextInt( bound: 4)+1;
        switch(count){
            case 1 :
                imageView.setImageResource(R.drawable.k_pic1);
                txtResult.setText("김치찌개");
                break;
            case 2 :
                imageView.setImageResource(R.drawable.k_pic2);
                txtResult.setText("된장찌개");
                break;
            case 3 :
                imageView.setImageResource(R.drawable.k_pic3);
                txtResult.setText("부대찌개");
                break;
            case 4 :
                imageView.setImageResource(R.drawable.k_pic4);
                txtResult.setText("감자탕");
                break;
        }
    }
});
```

Random유틸을 사용하여 4개의 데이터를 switch문을 통해 각 값을 나누고 count에 랜덤1-4까지중 하나를 랜덤 으로 넣어 해당하는 case문을 실행해 이미지와 text값을 출력한다.

라운드를 나누는 코드이다

```
layout = (LinearLayout) findViewById(R.id.roundback);
int food = getIntent().getIntExtra( name: "food", defaultValue: 0);
int round = getIntent().getIntExtra( name: "round", defaultValue: 0);

if (food == 1 && round == 8) {
    layout.setBackgroundResource(R.drawable.back_w_08); // 8강
} else if (food == 1 && round == 4) {
    layout.setBackgroundResource(R.drawable.back_w_04); // 4강
} else if (food == 1 && round == 2) {
    layout.setBackgroundResource(R.drawable.back_0); // final
} else if (food == 2 && round == 8) {
    layout.setBackgroundResource(R.drawable.back_w_08); // 8강
} else if (food == 2 && round == 4) {
    layout.setBackgroundResource(R.drawable.back_w_04); // 4강
} else if (food == 2 && round == 2) {
    layout.setBackgroundResource(R.drawable.back_0); // final
} else if (food == 3 && round == 8) {
    layout.setBackgroundResource(R.drawable.back_w_08); // 8강
} else if (food == 3 && round == 4) {
    layout.setBackgroundResource(R.drawable.back_w_04); // 4강
} else if (food == 3 && round == 2) {
    layout.setBackgroundResource(R.drawable.back_0); // final
} else if (food == 4 && round == 8) {
    layout.setBackgroundResource(R.drawable.back_w_08); // 8강
} else if (food == 4 && round == 4) {
    layout.setBackgroundResource(R.drawable.back_w_04); // 4강
} else if (food == 4 && round == 2) {
    layout.setBackgroundResource(R.drawable.back_0); // final
} else if (food == 1 && round == 16) {
    layout.setBackgroundResource(R.drawable.back_w_16); // 16강
```



```

        Intent roundIntent = new Intent( packageContext: RoundActivity.this,
            GameActivity.class);
        roundIntent.putExtra( name: "food", value: 2);
        startActivity(roundIntent);
        finish();
        // overridePendingTransition(anim.fade_in, anim.fade_in);
        overridePendingTransition( enterAnim: 0, exitAnim: 0);
    }
}, delayMillis: 1500);
else if (food == 3 && round == 16) {
    layout.setBackgroundResource(R.drawable.back_w_16);|
    new Handler().postDelayed(new Runnable() {

```

```

    } else if (food == 4 && round == 16) {
        layout.setBackgroundResource(R.drawable.back_w_16);

        new Handler().postDelayed(new Runnable() {

            @Override
            public void run() {
                // TODO Auto-generated method stub

                Intent roundIntent = new Intent( packageContext: RoundActivity.this,
                    GameActivity.class);
                roundIntent.putExtra( name: "food", value: 4);
                startActivity(roundIntent);
                finish();
                // overridePendingTransition(anim.fade_in, anim.fade_in);
                overridePendingTransition( enterAnim: 0, exitAnim: 0);
            }
        }, delayMillis: 1500);
    }
}

```

16강부터 결승까지 라운드 배경이미지를 바꾸고 라운드가 넘어갈 때 데이터값을 게임 액티비티로 보내준다

```
foodImg1 = (ImageButton) findViewById(R.id.imageButton_foodimg1);
foodImg2 = (ImageButton) findViewById(R.id.imageButton_foodimg2);
food = getIntent().getIntExtra( name: "food",  defaultValue: 0);
if (food == 1) {
    GameManager.getInstance().initKgame();
    // GameManager.getInstance().inittmp();
    GameManager.getInstance().randomList();

} else if (food == 2) {
    GameManager.getInstance().initCgame();
    GameManager.getInstance().randomList();
} else if (food == 3) {
    GameManager.getInstance().initJgame();
    GameManager.getInstance().randomList();
} else if (food == 4) {
    GameManager.getInstance().initWgame();
    GameManager.getInstance().randomList();
} else if (food == 5) {
    GameManager.getInstance().initWgame();
    GameManager.getInstance().randomList();
}

gameCount = 0;
nowPlay = gameCount;
nowPlay2 = gameCount + 1;
nowPlay3 = gameCount + 2;
nowPlay4 = gameCount + 3;

foodImg1.setImageResource(GameManager.getInstance().fdwc.get(
    nowPlay).getImg());

foodImg2.setImageResource(GameManager.getInstance().fdwc.get(
    nowPlay2).getImg());

foodImg1.setOnClickListener(new OnClickListener() {
```

```

@Override
public void onClick(View v) {
    for (int i = 0; i < 9; i++) {
        new ParticleSystem( a: GameActivity.this, maxParticles: 2, starNumber[i],
            timeToLive: 1000)
            .setSpeedByComponentsRange( speedMinX: -0.2f, speedMaxX: 0.2f, speedMinY: -0.2f,
                speedMaxY: 0.04f).setAcceleration( acceleration: 0.000009f, angle: 90)
            .setInitialRotationRange(0, 360)
            .setRotationSpeed(120).setFadeOut(150)
            .addModifier(new ScaleModifier( initialValue: 0f, finalValue: 1.5f, startMillis: 0, endMillis: 500))
            .oneShot(v, numParticles: 20);
    }

    new Handler().postDelayed(new Runnable() {

        @Override
        public void run() {

            // TODO Auto-generated method stub
            if (gameCount == 7) {

                Intent roundIntent = new Intent( packageContext: GameActivity.this,
                    RoundActivity.class);
                roundIntent.putExtra( name: "food", food);
                roundIntent.putExtra( name: "round", value: 8);
                startActivity(roundIntent);
                overridePendingTransition( enterAnim: 0, exitAnim: 0);

                new Handler().postDelayed(new Runnable() {

                    @Override
                    public void run() {
                        // TODO Auto-generated method stub

```

```

@Override
public void run() {
    // TODO Auto-generated method stub

    GameManager.getInstance().fdwc
        .remove(nowPlay2);
    // GameManager.getInstance().randomList();
    nowPlay = 0;
    nowPlay2 = 1;
    foodImg1.setImageResource(GameManager
        .getInstance().fdwc.get(
            nowPlay).getImg());
    foodImg2.setImageResource(GameManager
        .getInstance().fdwc.get(
            nowPlay2).getImg());
}
}, delayMillis: 1500);

```

게임메니저 액티비에서 데이터 값들을 받아오고 선택되었을 때 나타는 효과가 생기며 라운드마다 선택된 데이터들을 다음 라운드로 다시 보내고 결승에서 뽑힌 데이터는 FinishActivity로 보내게 된다.

```

@Override
public void onFocusChanged(boolean hasFocus) {
    if (hasFocus && !animInitialized) {
        for (int i = 0; i < 12; i++) {
            new ParticleSystem( a: this, maxParticles: 10, pollenNumber[i], timeToLive: 10000)
                .setSpeedModuleAndAngleRange( speedMin: 0f, speedMax: 0.1f, minAngle: 180, maxAngle: 180)
                .setRotationSpeed(144).setAcceleration( acceleration: 0.000025f, angle: 90)
                .emit(findViewById(imageNumber[i]), particlesPerSecond: 1);
        }
        this.animInitialized = true;
    }
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_finish);
    database= FirebaseDatabase.getInstance();
    myRef= database.getReference();

    imgProfile = (ImageView) findViewById(R.id.imageView_profile);
    tv_name = (TextView) findViewById(R.id.textView_name);
    String name=getIntent().getStringExtra( name: "name");
    int profile=getIntent().getIntExtra( name: "img", defaultValue: 0);
    imgProfile.setImageResource(profile);
    tv_name.setText(name);
    String data = tv_name.getText().toString();
}

```

```

imgProfile = (ImageView) findViewById(R.id.imageView_profile);
tv_name = (TextView) findViewById(R.id.textView_name);
String name=getIntent().getStringExtra( name: "name");
int profile=getIntent().getIntExtra( name: "img", defaultValue: 0);
imgProfile.setImageResource(profile);
tv_name.setText(name);
String data = tv_name.getText().toString();
count++;
btn_rank = (Button)findViewById(R.id.rank);
btn_rank.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(getApplicationContext(),Rank.class);
        startActivity(intent);
    }
}

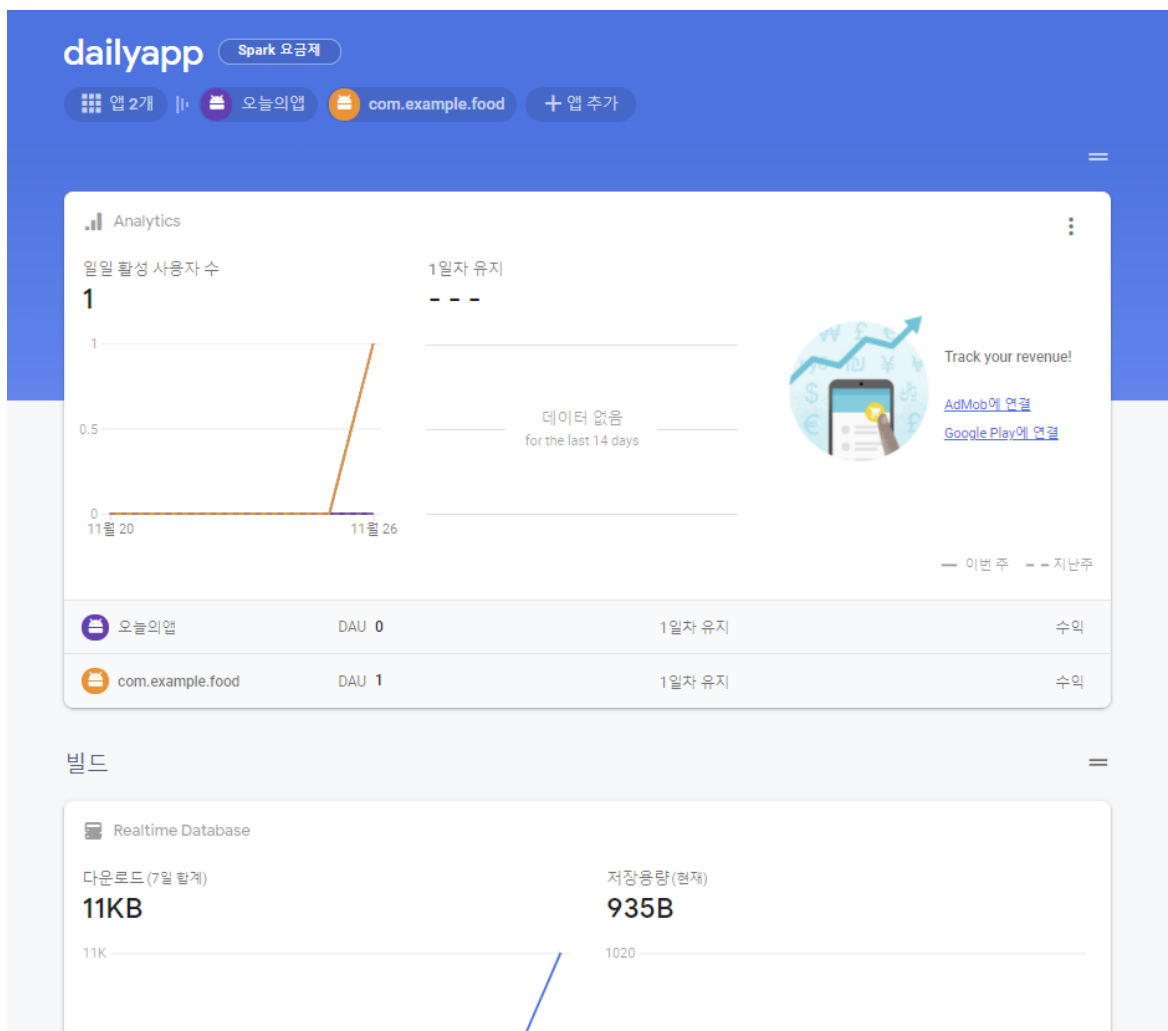
```

최종적으로 결승에서 뽑힌 데이터의 이미지와 텍스트가 출력되는 화면의 코드로 최종라운드에서 선택된 이미지와 텍스트를 중앙에 띄우고 그 아래 랭킹 보기 버튼을 만들어 최종 결과들의 랭크를 볼 수있다(구현은 하지 못했음)

다음으로는 Firebase인데 Firebase는 로그인과 데이터 주고받기를 하기 쉽게 도와주기 때문에 선택하게 되었다.

```
implementation 'com.google.firebase:firebase-auth:19.2.0'
implementation 'com.google.firebase:firebase-database:20.0.2'
implementation 'com.google.firebase:firebase-storage:20.0.0'
```

Firebase를 gradle에 implement 해주고 Firebase 홈페이지에서 앱을 프로젝트를 추가하고



안드로이드 스튜디오와 연동을 시킨다 .구글-서비스 JSON파일을 받아 디렉토리에 추가하고 앱 등록을 통해 안드로이드와 연결시킨다 그리고 동기화를 시키면 완료된다

그 후 로그인을 위해 authentication 에서 로그인 방법을 추가하고 인증 라이브러리를 추가하고

```
private FirebaseAuth mAuth = null;
private GoogleSignInClient mGoogleSignInClient;
private static final int RC_SIGN_IN = 9001;
private SignInButton signInButton;
signInButton = findViewById(R.id.signInButton);
mAuth = FirebaseAuth.getInstance();
GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(R.string.default_web_client
_id))
    .requestEmail()
    .build();
mGoogleSignInClient = GoogleSignIn.getClient(this, gso);
if (mAuth.getCurrentUser() != null) {
    Intent intent = new Intent(getApplication(),
AfterActivity.class);
    startActivity(intent);
    finish();
}
private void signIn() {
    Intent signInIntent = mGoogleSignInClient.getSignInIntent();
    startActivityForResult(signInIntent, RC_SIGN_IN);
}
@Override
public void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    // Result returned from launching the Intent from
GoogleSignInApi.getSignInIntent(...);
    if (requestCode == RC_SIGN_IN) {
        Task<GoogleSignInAccount> task =
GoogleSignIn.getSignedInAccountFromIntent(data);
        try {
            // Google Sign In was successful, authenticate with
Firebase
            GoogleSignInAccount account =
task.getResult(ApiException.class);
            firebaseAuthWithGoogle(account);
        } catch (ApiException e) {
        }
    }
}
```

```

    }
}
private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {

    AuthCredential credential =
    GoogleAuthProvider.getCredential(acct.getIdToken(), null);
    mAuth.signInWithCredential(credential)
        .addOnCompleteListener(this, new
    OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult>
task) {
            if (task.isSuccessful()) {
                // Sign in success, update UI with the
signed-in user's information

                Snackbar.make(findViewById(R.id.layout_main), "Authentication
                Succeeded.", Snackbar.LENGTH_SHORT).show();
                FirebaseUser user =
                mAuth.getCurrentUser();
                updateUI(user);
            } else {
                // If sign in fails, display a message to
the user.

                Snackbar.make(findViewById(R.id.layout_main), "Authentication
                Failed.", Snackbar.LENGTH_SHORT).show();
                updateUI(null);
            }
        }
    });
}

private void updateUI(FirebaseUser user) { //update ui code here
    if (user != null) {
        Intent intent = new Intent(this, AfterActivity.class);
        startActivity(intent);
        finish();
    }
}

mAuth = FirebaseAuth.getInstance();
private void signOut() {
    FirebaseAuth.getInstance().signOut();
}

```

구글 문서에 나와있는 코드를 조금 바꿔서 로그인기능을 사용하였다.

졸업작품을 진행하면서 공부한 내용

졸업 작품을 진행하면서 공부했던 것들은 안드로이드 스튜디오, Firebase, AWS

서버 가 있다. 주로 안드로이드 스튜디오 위주로 공부했고 서버를 맡았기 때문에

AWS를 초반에 공부하다가 오늘의 앱이 무산되고 새로 오늘 뭐먹지를

하게 되어서 조금 사용하기 쉬운 Firebase를 사용하게 되어 Firebase관련하여서도 공부를 잠깐 했다.

안드로이드 스튜디오는 많은 기능들을 사용할 수 있다는 장점이 있었지만 에뮬레이터가 오류가 너무 많고 실행 속도도 너무 느려 답답했던 적이 많았다. 기회가 된다면 오늘 뭐먹지 같은 앱은 React Native같은 것으로 만드는 것이 더 편할 것 같다는 생각이 들었다.

졸업작품 완료 후 느낀점

졸업작품을 만들고 나서 느낀 점은 일단 아이디어 선정이 제일 중요하고 계획부분이 정말 중요하다는 것을 많이 느꼈다. 그리고 협업하는 것이 쉽지 않다는 것도 느꼈다.

비록 이번 졸업작품에서 뛰어나거나 인상 깊은 작품을 만들지는 못했지만 그래도 실패하는 과정 속에서 스토리보드, 계획 프로젝트의 플로우 차트 같은 것들이 정말 중요하다는 것을 깨달았고 팀원들끼리 의견을 조율하고 개발 파트를 나누면서 개발해 나가는 것을 부족하지만 어느정도 알 수 있었던 시간이었던 거 같다. 개발직이 된다면 졸업작품을 통해 느낀 점을 토대로 단점을 보완하고 장점을 많이 발전시킬 수 있게 되었으면 좋겠다.