



Kookmin University
College of Software
Computer Science

Software Project I

AD Project

과제명	AD Project
소속	<i>Kookmin University Computer Science</i>
학번	<i>20191675 / 20191683</i>
성명	<i>조현진 / 최지훈</i>
문서 제목	AD PROJECT - EARTHQUAKE
교수	최진우


Version	1.0
Date	2019.06.17

 국민대학교 컴퓨터공학부 소프트웨어프로젝트I			
	과제명	AD Project	
	학번/이름	20191675 / 조현진, 20191683 / 최지훈	
	Confidential Restricted	Version 1.0	

문서 정보 / 수정 내역

Filename	AD Project
원안작성자	Ji-hun choi / hyun-jin jo
수정작업자	Ji-hun choi / hyun-jin jo

수정날짜	작성자	Revision	추가/수정 항목	내 용
2019.06.17	Ji-hun choi	1.0	First Write	First Write
2019.06.19	Ji-hun choi	2.0	Last Write	Last Write

 국민대학교 컴퓨터공학부 소프트웨어프로젝트I			
	과제명	AD Project	
	학번/이름	20191675 / 조현진, 20191683 / 최지훈	
	Confidential Restricted	Version 1.0	

목 차

1	개요	4
1.1	프로젝트 개요	4
1.2	추진 배경 및 필요성	4
2	개발 내용 및 결과물	4
2.1	목표	4
2.2	연구/개발 내용 및 결과물	4
2.2.1	서버 알고리즘	4
2.2.2	클라이언트(앱 인벤터) 알고리즘	7
2.2.3	현실적 제한 요소 및 그 해결 방안	18
2.2.4	결과물 화면	19
3	자기평가	21
4	참고 문헌	21
	프로젝트 준비하는 것과 발표하는 것 정말 재미있었습니다!!!	21

 국민대학교 컴퓨터공학부 소프트웨어프로젝트I		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0

1 개요

프로젝트의 전 과정을 설명한다.

1.1 프로젝트 개요

프로젝트를 하는 이유, 개발 내용, 결과물, 등을 나열한다.

1.2 추진 배경 및 필요성

지금까지 소프트웨어 프로젝트 1에서 배운 내용을 바탕으로 간단한 프로젝트를 진행하고자 한다. 프로젝트를 진행하면서 여태 배운 내용을 복습하고 또한 진행에 있어서 필요한 내용을 새로 공부한다. 무엇보다도 한 번도 해본적이 없는 팀프로젝트를 경험해보고 발표도 한다. 또한 문서 작성 스킬도 업그레이드한다.

2 개발 내용 및 결과물

개발 내용과 결과물을 보여준다.

2.1 목표

서버 클라이언트 구조를 활용하여 간단한 게임을 만든다. Python 을 이용하여 Json 형식으로 요청에 응답하는 서버를 만든다. 또한 게임을 할 클라이언트는 App Inventor 로 구현한다.

2.2 연구/개발 내용 및 결과물

게임적인 요소를 더하기 위하여 로그인, 랭킹, 상점, 설정 등을 구현하였다.

2.2.1 서버 알고리즘

서버인 파이썬 코드와 클라이언트인 앱 인벤터를 나눠서 설명할 것이다.

```

1 from wsgiref.simple_server import make_server
2 from cgi import parse_qs, escape
3 import json
4 import pickle

```



```
5
6 #load_stored_score_board
7 f = open('score_data.dat', 'rb')
8 score_board = pickle.load(f)
9 f.close()
10
11 def application(envIRON, start_response):
12     d = parse_qs(envIRON['QUERY_STRING'])
13
14     user_id = escape(d.get('id', [''])[0])
15     user_score = int(escape(d.get('score', [''])[0]))
16     signal = int(escape(d.get('signal', [''])[0]))
17     code = 0
```

Colored by Color Scripter

이 파이썬 파일은 score_data.dat 파일과 같은 위치에 있다. 스코어보드의 정보를 저장하고 불러오기 위해 pickle을 사용했다. 또한 해당 API주소의 query에는 Id, score, signal을 포함한다.

코드는 query에 포함된 signal에 따라 실행된다. Signal이 1일 때 실행되는 코드는 로그인을 위해 중복확인 버튼을 눌렀을 때만 실행된다.

```
1 #id dupliate check
2 if signal == 1:
3     if user_id in score_board:
4         code = 'T'
5         response_body = json.dumps({'ID': user_id, 'SCORE': score_board[user_id],
6                                     'BOARD': score_board, 'code': code})
7     else:
8         code = 'F'
9         score_board[user_id] = 0
10    response_body = json.dumps({'ID': user_id, 'SCORE': score_board[user_id],
```

CS


 국민대학교 컴퓨터공학부 소프트웨어프로젝트I		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0

```
11 | 'BOARD': score_board, 'code': code}}
```

만약 query에 포함된 id가 이미 score_board에 있다면 그것은 중복이므로 code에 T를 실어서 보낸다. 만약 없다면 사용이 가능한 아이디 이므로 해당 유저의 아이디를 score_board에 0점으로 등록하고, code에 F를 실어서 보낸다.


또한 signal이 2번인 경우는 점수등록을 위해 점수등록 버튼을 눌렀을 때이다.

```
1 #score_register
2 if signal == 2:
3     if score_board[user_id] > user_score:
4         score_board[user_id] = score_board[user_id]
5         code = 'S'
6     else:
7         score_board[user_id] = user_score
8         code = 'S'
```

Colored by Color Scripter 

점수등록을 위한 코드이면서 최고점수만을 등록하는 코드이다. 만약 현재 요청한 점수가 기존의 점수보다 높다면 새로운 점수를 등록하고, 낮다면 기존의 점수를 등록한다.

```
1 #load_score_board or reset
2 sorting_score_board = sorted(score_board.items(), key = lambda x:x[1], reverse=True)
3 response_body = json.dumps({'ID': user_id, 'SCORE': score_board[user_id],
4                             'BOARD': sorting_score_board, 'code': code})
5 status = '200 OK'
6 response_headers = [('Content-Type', 'application/json'),('Content-
Length', str(len(response_body)))]
```



헤더를 만들고, signal이 1번과 2번이 아닌 경우, 즉 단순히 score_board만 보길 원하는 경우의 코드이다. 헤더를 만드는 것은 꼭 필요하기 때문에 해당코드는 사실 항상 실행된다. 또한 랭킹을 점수순으로 정렬하기 위해 딕셔너리를 새로운 리스트로 정렬하여 만든다.

```
1 #store score_board
```





과제명	AD Project	
학번/이름	20191675 / 조현진, 20191683 / 최지훈	
Confidential Restricted	Version 1.0	

```

2 f = open('score_data.dat', 'wb+')
3 start_response(status, response_headers)
4 pickle.dump(score_board, f)
5 f.close()
6
7 return [response_body]
8
9 httpd = make_server('192.168.200.142', 8051, application)
10 httpd.serve_forever()
11

```

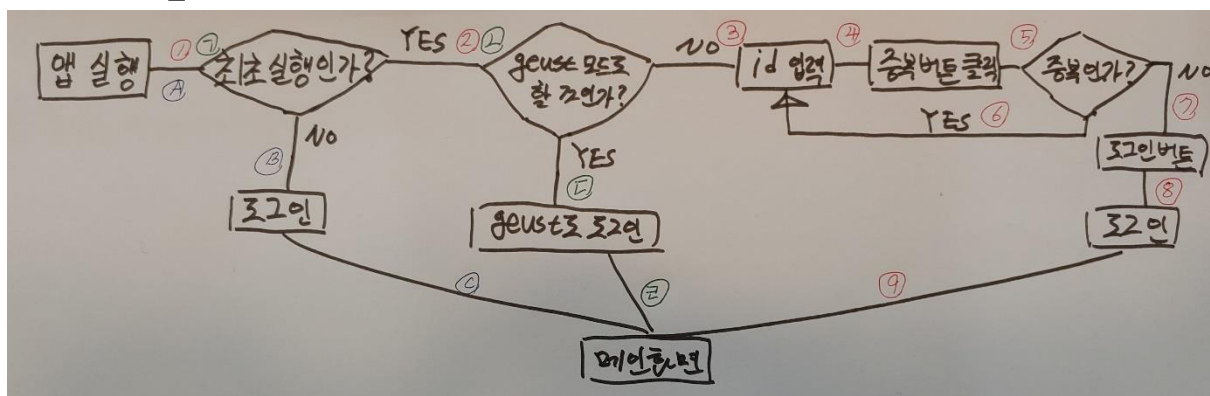
Colored by Color Scriptor

마지막으로 수정된 score_board를 피클링을 통해 저장하고, response_body를 반환한다.

2.2.2 클라이언트(앱 인벤터) 알고리즘

이제 게임의 핵심요소인 로그인, 게임, 순위표, 상점의 알고리즘에 대해서 설명한다.

- 로그인




로그인을 할 때 세 가지의 경우가 있다. 최초접속이 아닌 기존에 접속한 이력이 있는 경우, 게스트 모드로 로그인한 경우, 새로운 아이디를 만들어서 로그인하는 경우로 총 3가지의 경우가 있다.

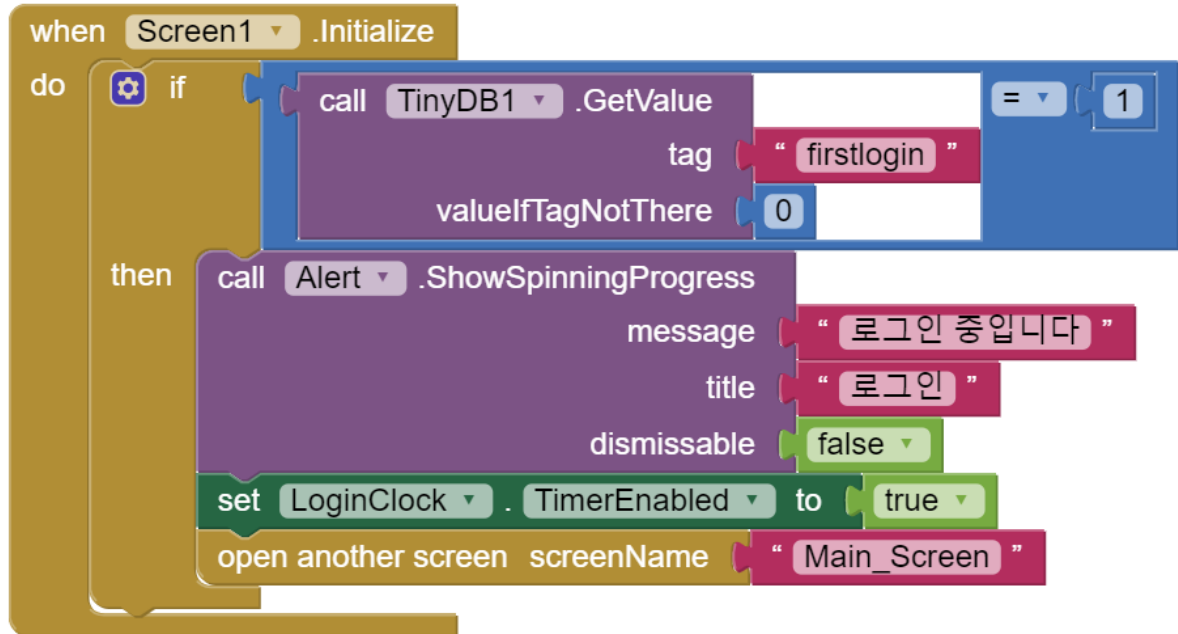
1. 최초 접속이 아닌 경우에 대해서 설명하겠다.

위의 그림에서 A, B, C에 해당한다.

A- B- C 앱 실행과 동시에 Tiny DB에 저장된 firstlogin의 값이 0(최초실행)인지 1(실행이력 O)

 국민대학교 컴퓨터공학부 소프트웨어프로젝트I		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0

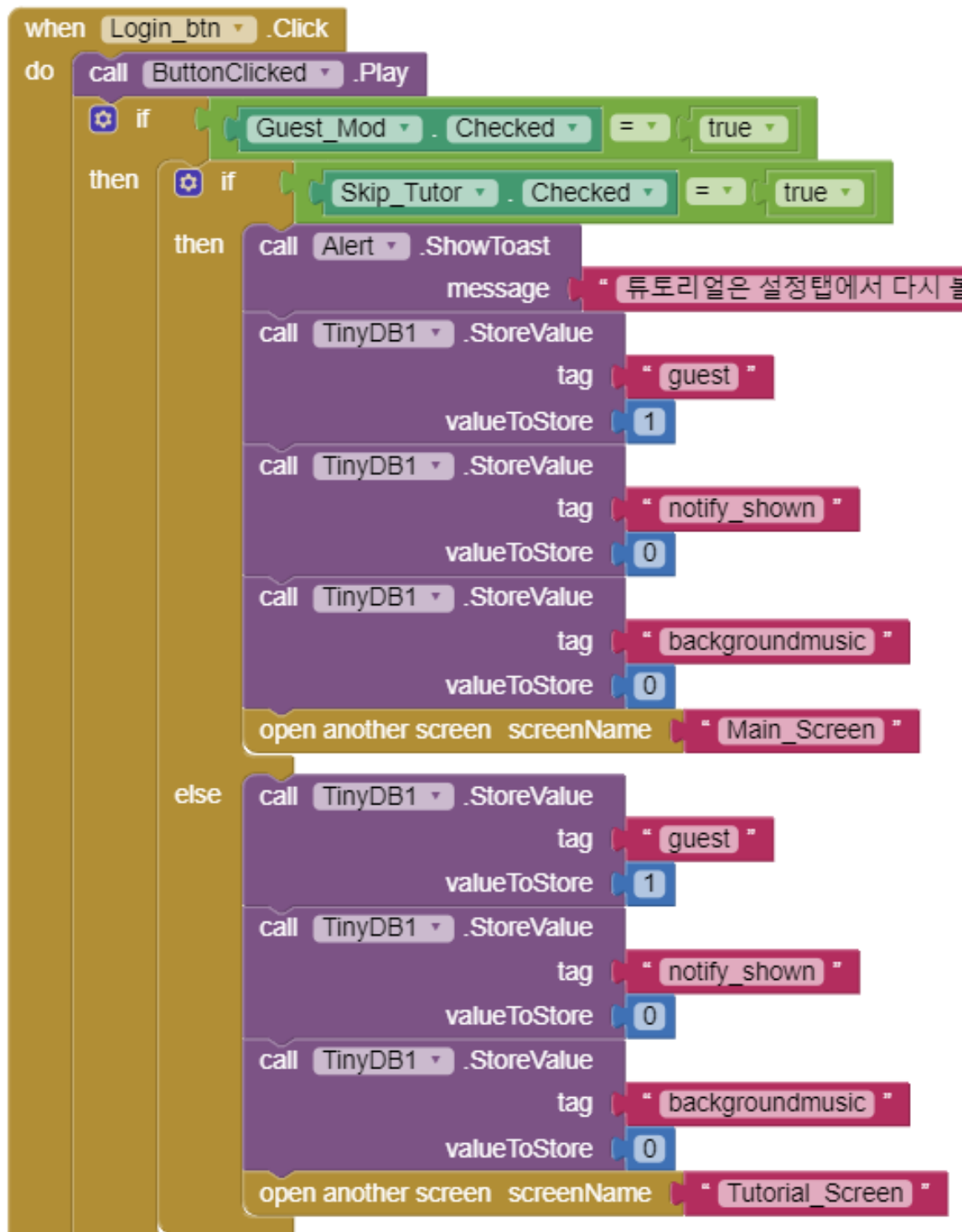
인지 확인하고 만약 로그인 이력이 있다면 바로 로그인을 하고 메인 화면으로 이동한다.




2. 게스트 모드로 로그인 할 때

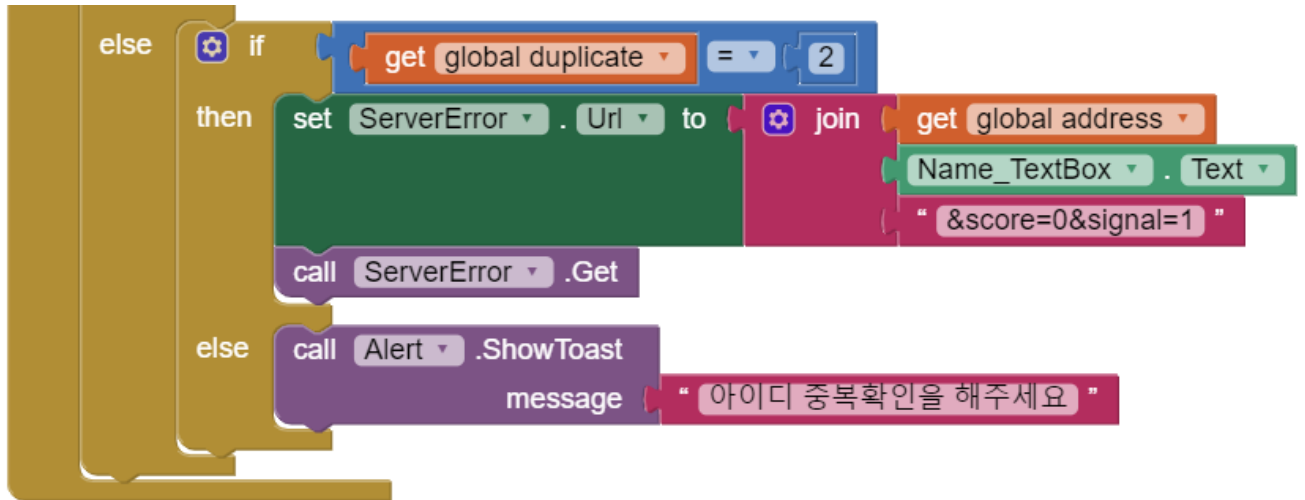
위의 그림에서 ㄱ, ㄴ, ㄷ, ㄹ에 해당한다.

이 경우도 위의 1번 경우와 마찬가지로 앱 실행과 동시에 최초 접속인지 확인한다. 하지만 로그인 이력이 없으니 id등록을 진행해야 하는데, 서버가 켜져 있지 않거나, 인터넷이 접속되지 않은 경우에는 id중복체크 확인을 할 수 없으므로 게스트모드로 접속해야 한다.



만약 게스트모드로 로그인하고자 할 경우 게스트모드 체크박스에 체크를 한다. 또한 튜토리얼을 원치 않을 경우에도 튜토리얼 스킵 체크박스에 체크를 한다. 이후 로그인 버튼을 누르면 해당 유저가 게스트인지 아닌지를 판단하는 변수 geust에 1(게스트이다. 0은 게스트가 아니다.)을 대입한다. 이 변수는 향후 자신의 점수를 랭킹에 올릴 수 있는지 판단하는데 사용한다. 튜토리얼을 원하면 튜토리얼 화면을 먼저 켜고, 그렇지 않다면 메인 화면으로 연결된다.

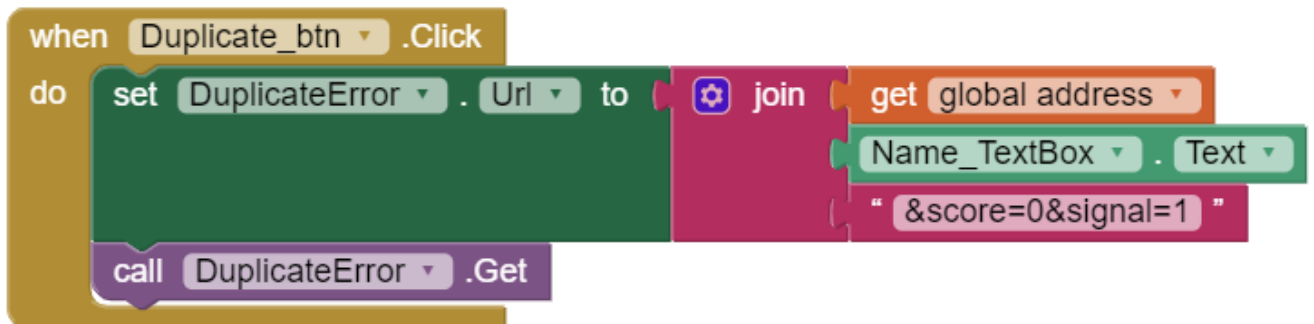
 국민대학교 컴퓨터공학부 소프트웨어프로젝트I		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0



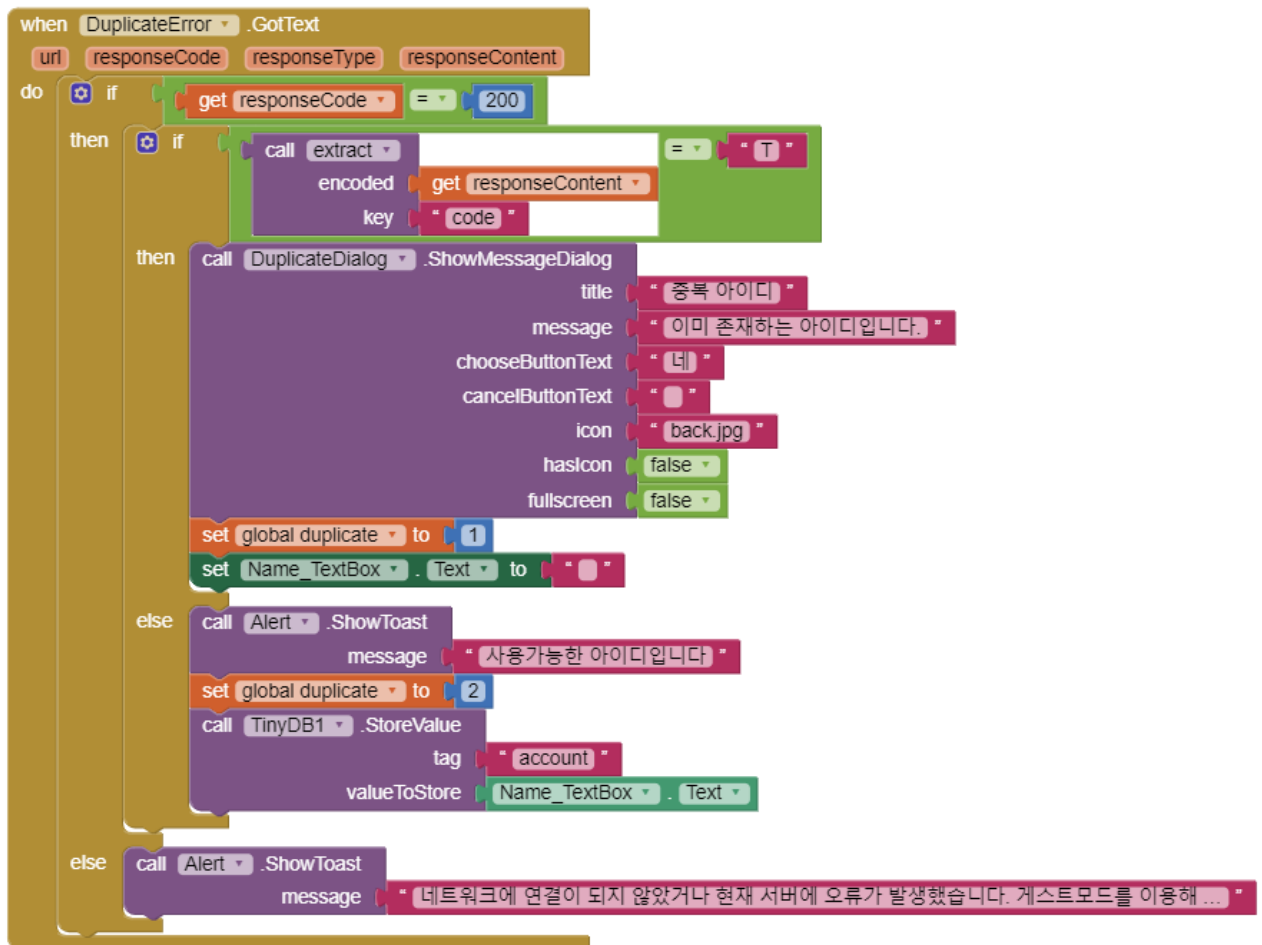
이 코드는 위의 코드에 연결 되어있다. 로그인 버튼을 눌렀을 때, 게스트모드가 아니고 중복확인을 끝냈다면, 서버의 스코어보드에 해당 id로 0점을 등록한다. 만약 중복확인을 끝내지 않았다면 (duplicate가 2가 아니라면) 아이디 중복확인을 하라는 토스트를 띄운다.

3. 새로운 id를 만들어서 로그인하는 경우

1과 2의 경우와 같이 처음에는 앱을 실행하자마자 최초 로그인 인지 판단하고, 아니라면 아이디를 입력하고 중복확인버튼을 클릭한다.



서버로 보낼 URL을 만들고 signal은 1번을 실어서 보낸다. Signal 1번은 서버에서 중복확인을 시도한다는 것을 알려주는 신호이다. 신호를 보내서 서버에서 json으로 받아온다.



responseCode가 200이고, key값인 code가 T라면(서버에서 중복이면 code를 T로 설정한다.) 중복이므로 다이얼로그를 띄워주고, 중복임을 판단하는 duplicate변수에 1을 할당한다. 중복이 아닐 때까지 중복확인을 계속하고, 만약 중복이 아니라면 사용이 가능하다는 토스트를 띄우고, duplicate 변수에 2를 할당한다음에 사용자의 계정을 TinyDB에 등록한다.

 국민대학교 컴퓨터공학부 소프트웨어프로젝트I		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0

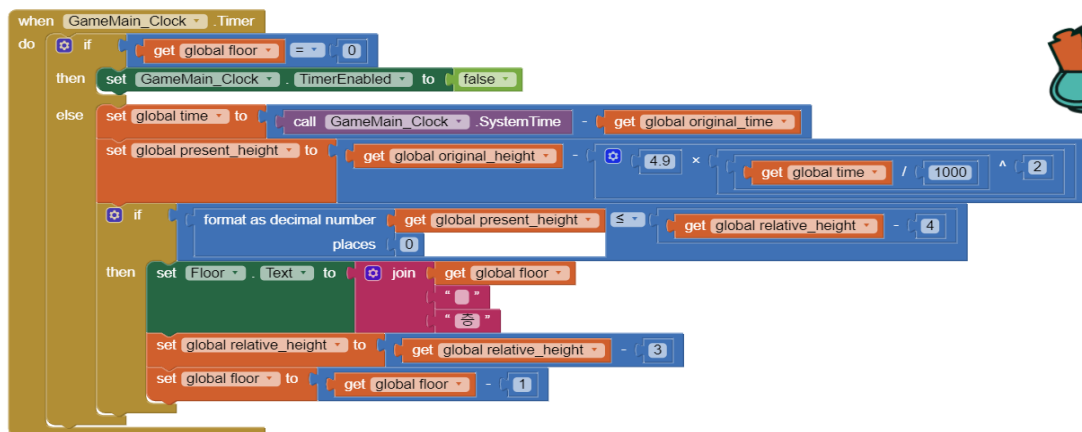
- 게임

우선 실질적인 게임 플레이를 담당하는 전체의 코드는 이러하다.



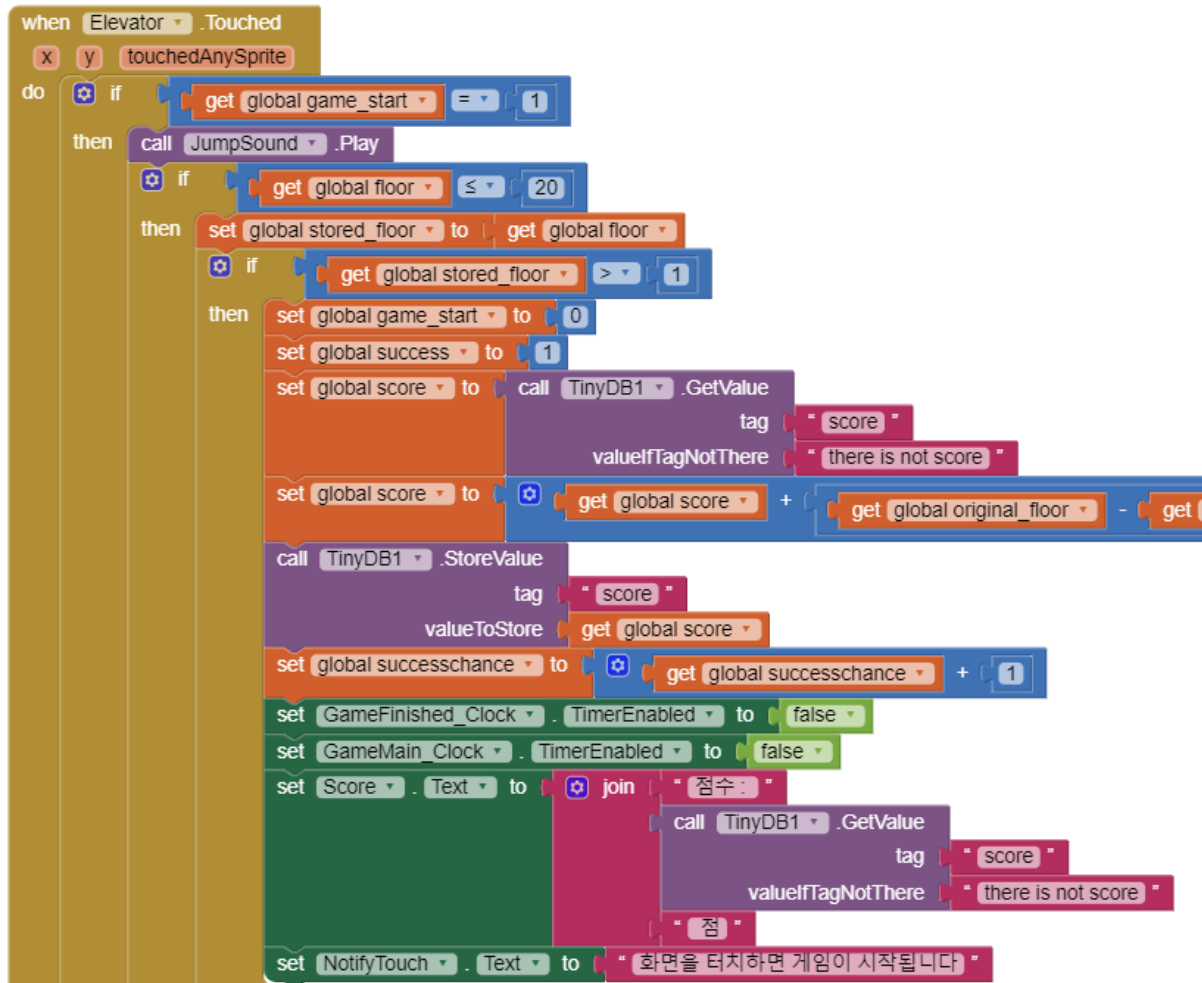
굉장히 복잡해서 핵심부분만 짚고 넘어가겠다.

게임의 층수를 계산하는 메인 클락의 코드이다

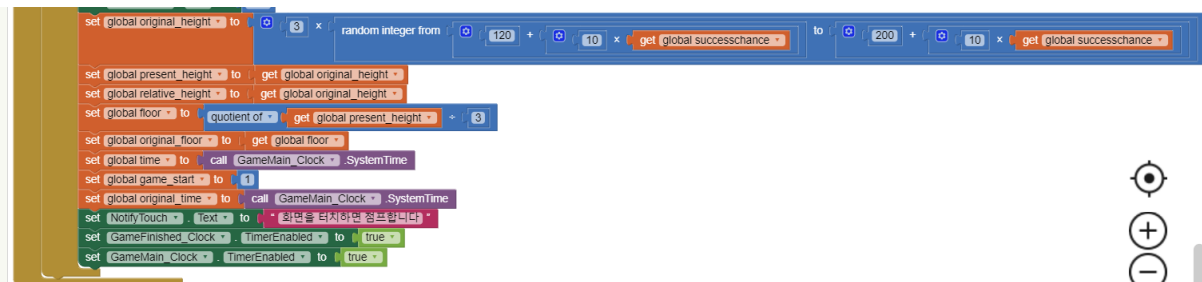




우선 시간을 0초부터 흘러가게 하기위해 시간의 차를 구해서 공식에 대입하여 계산한다. 계산한 값이 소수점이 나오기 때문에 =연산자는 사용할 수 없고 부등호를 사용하였다. 이 클락은 게임 화면을 터치한 순간부터 다시 터치한 순간까지 계속 동작한다.



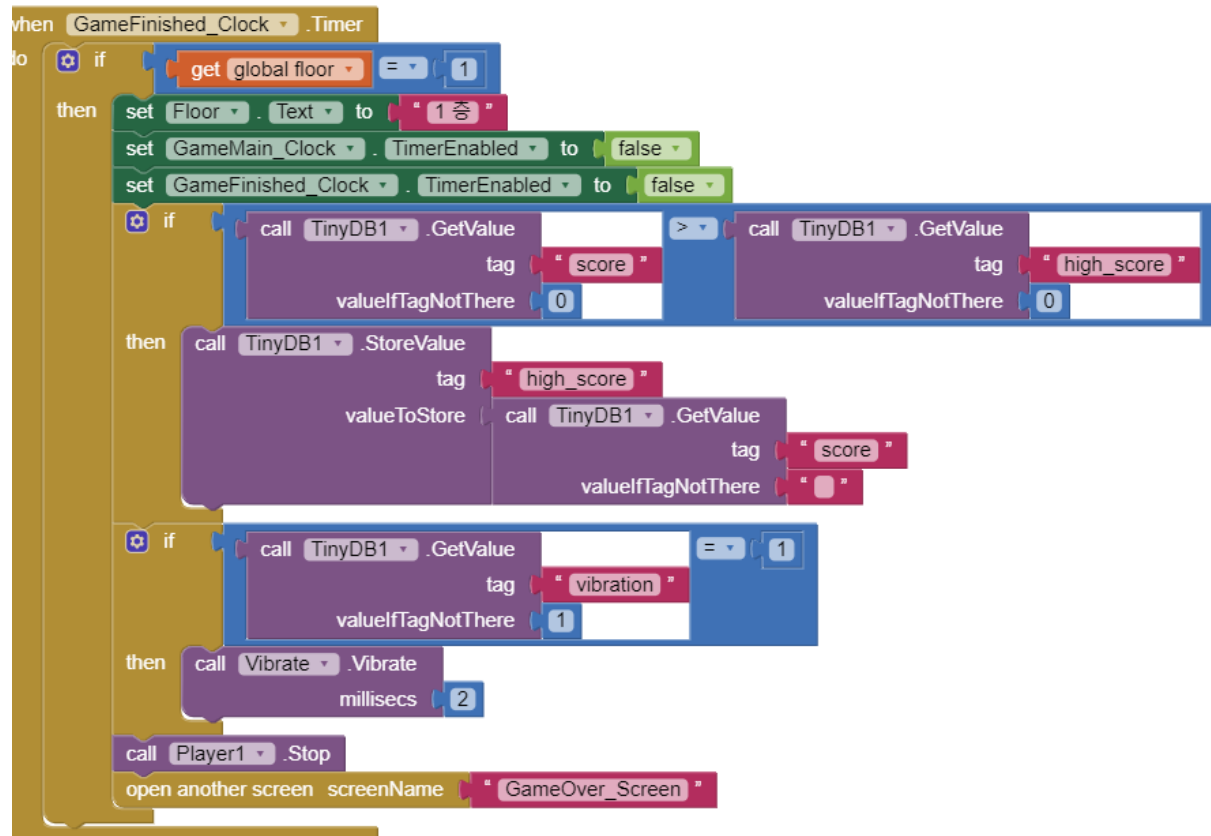
이 코드는 게임 중에 화면을 터치하여 캐릭터를 점프 시키는 부분이다. 터치를 한 시점이 20층보다 작고 1층보다 크면 성공한 것이므로 점수를 계산하고 TinyDB에 저장한다. 또한 게임을 하면 할수록 난이도가 어려워져야 재미있으므로



게임을 성공할 횟수를 기록하며 층수를 랜덤으로 높여간다.

 국민대학교 컴퓨터공학부 소프트웨어프로젝트I		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0

또한 엘리베이터가 1층에 도착을 하였는데도 점프를 하지 않았다면 실패한 것이니 예외를 처리할 코드도 작성한다.

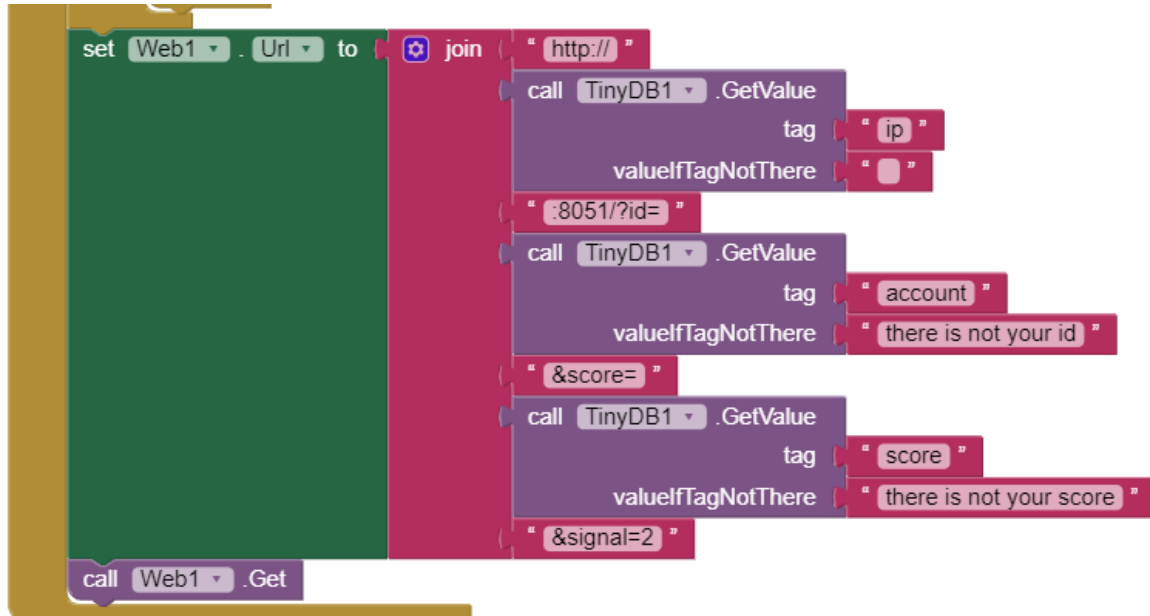


만약 1층에 도착한 경우에 게임오버 이므로 현재까지의 점수를 TinyDB에 저장하고 게임오버 스크린을 연다.

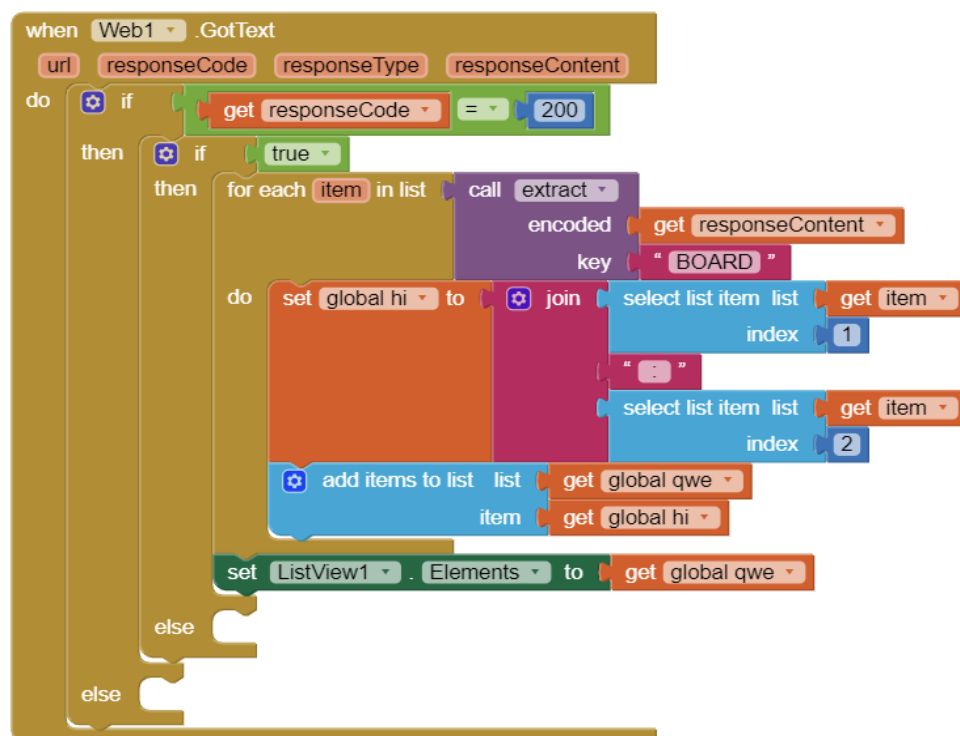
 국민대학교 컴퓨터공학부 소프트웨어프로젝트I		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0

- 순위표


게임을 할 때 상대방과 경쟁을 하면 더 재미있다. 따라서 서버를 이용하여 순위표를 만들었다.



이 코드는 순위표 화면이 켜지자마자 실행되는 부분이다. 바로 signal2로 주소를 만들어서 서버에 요청을 보낸다.



Web1이 요청을 받았을 때 2차원 배열로 된 값을 1차원 배열로 만들어서 리스트뷰에 할당하여 화면에 표시하는 코드이다

 국민대학교 컴퓨터공학부 소프트웨어프로젝트I		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0

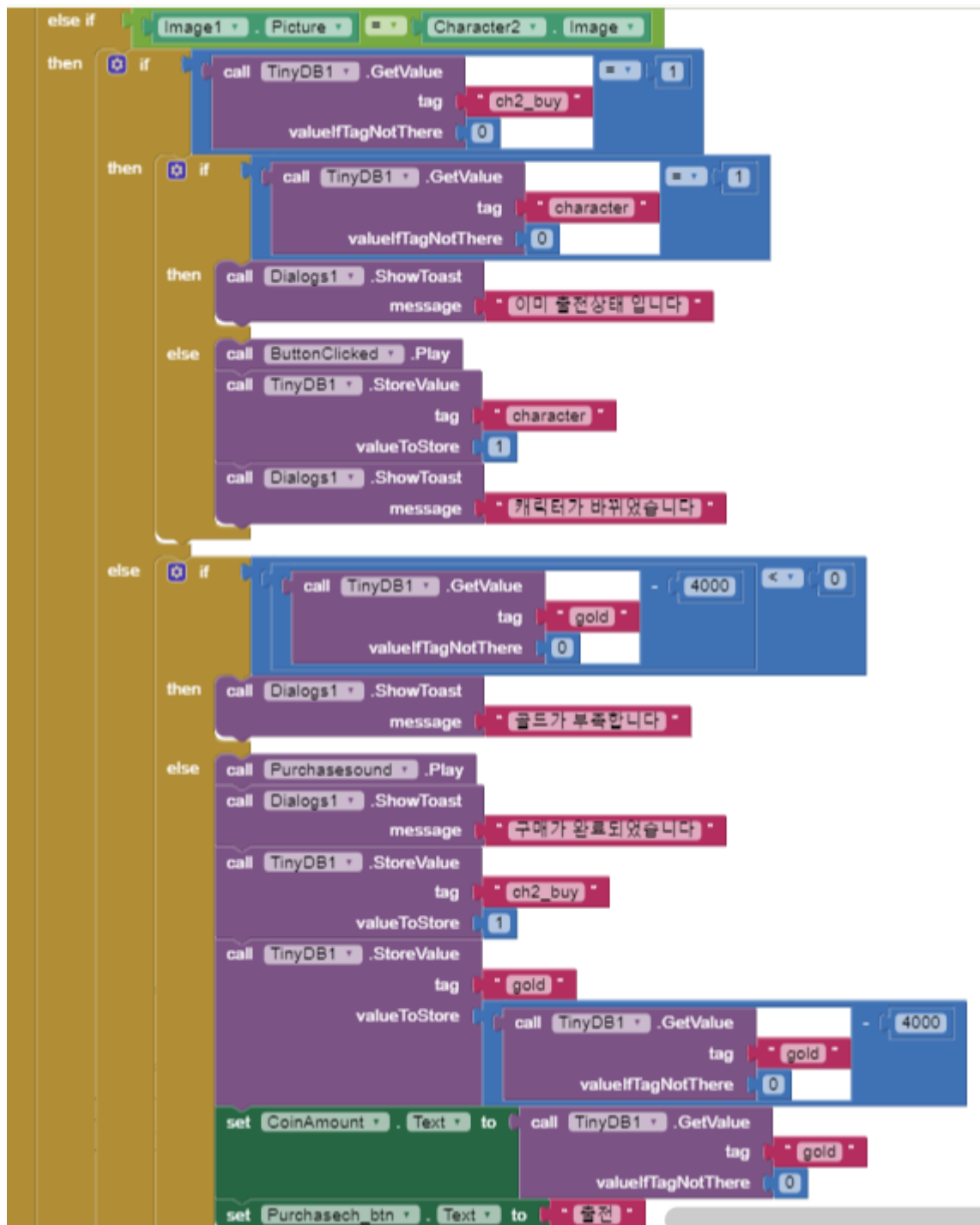
- 상점

이 코드는 상점코드의 전부이다.



핵심부분만 살펴보면 다음과 같다.

상점에서 캐릭터를 사고 해당 캐릭터로 이미지를 변경하는 코드이다.

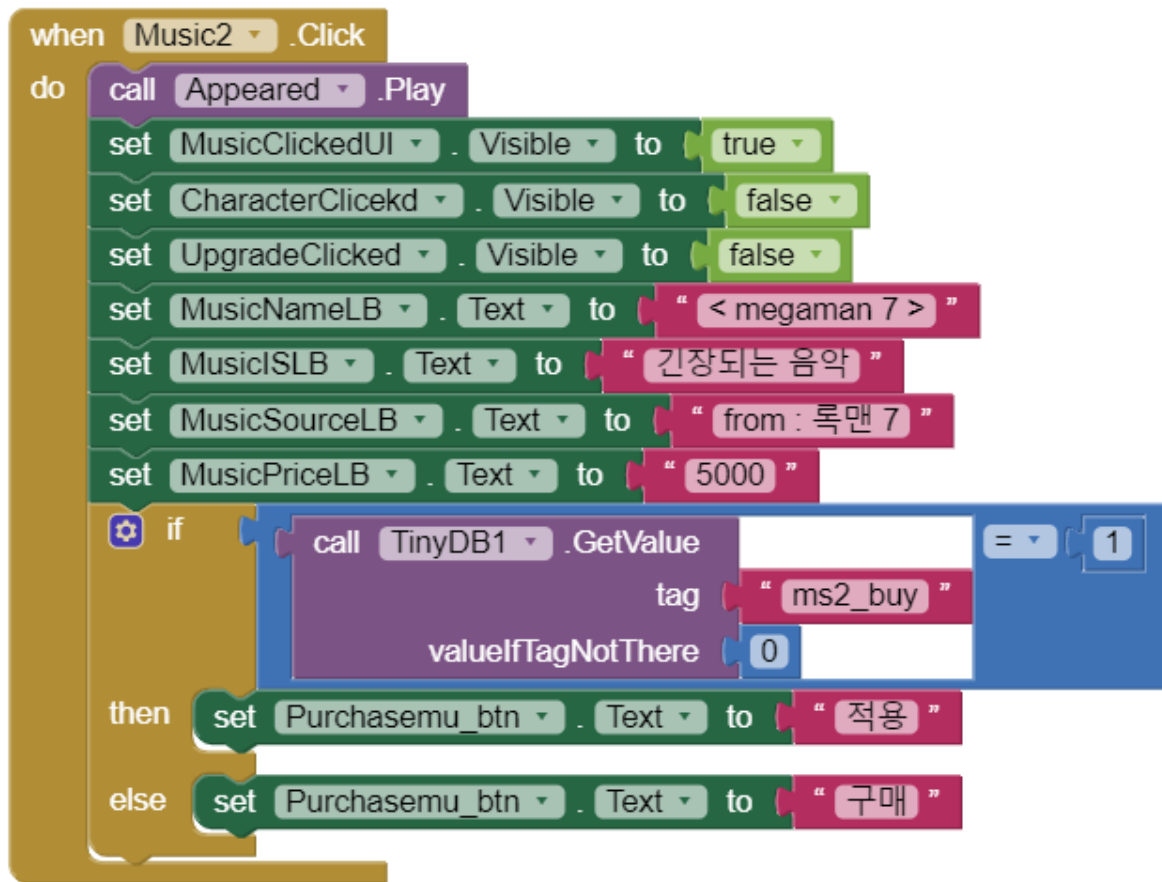


캐릭터를 구매하고, 화면에 캐릭터를 표시할 변수를 TinyDB에 저장한다. 만약 이미 해당 캐릭터로 설정되어 있다면 이미 출전상태라는 토스트를 띄운다. 또한 돈이 부족할 때와 구매했을 때 돈이 차감되는 것도 구현 되어있다.

또한 음악의 경우에도




과제명		AD Project	
학번/이름		20191675 / 조현진, 20191683 / 최지훈	
Confidential Restricted		Version 1.0	



해당음악을 산 적이 있는지 없는지 0과 1로 구분 지어 TinyDB에 저장하고 그 값에 따라 적용을 할지 구매를 할지 표시를 한다.


2.2.3 현실적 제한 요소 및 그 해결 방안

게임을 할 때 서버가 켜져 있지 않다면 서버에 점수를 올리거나 순위표를 불러올 수 없다. 따라서 항상 켜져 있는 서버를 구현만 한다면 상당히 완성도 있는 게임이 될 것이다.

 <div> <p>국민대학교</p> <p>컴퓨터공학부</p> <p>소프트웨어프로젝트I</p> </div>		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0

2.2.4 결과물 화면



 국민대학교 컴퓨터공학부 소프트웨어프로젝트I		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0

12:39 U+

Back

63

< 설정 >

소리

사운드 테스트

진동

ip 전환

ip주소 입력

전환

제작자

튜토리얼

문의하기

12:39 U+

Back

63

< 도전과제 >

< 엘리베이터 입문 >

게임 실행 - 500 골드

도전

< 감사의 보답 >

공유하기 - 100 골드

도전

후원하기 누르기 - 500 골드

도전

< 끝판왕 도전 >

최고기록 1000점 달성 - 1000 골드

도전

최고기록 5000점 달성 - 5000 골드

도전

최고기록 10000점 달성 - 10000 골드

도전

< ??? >

비밀의 문 열기 - 9999999 골드

도전

 국민대학교 컴퓨터공학부 소프트웨어프로젝트I		
	과제명	AD Project
	학번/이름	20191675 / 조현진, 20191683 / 최지훈
	Confidential Restricted	Version 1.0

3 자기평가

솔직히 서버도 완벽히 모르겠고, 앱 인벤터도 모든 기능을 완벽히 다루지는 못 한다. 하지만 이번프로젝트를 통해서 머릿속으로 생각한 것을 우리가 배운 것으로 만들 수 있다는 것을 깨닫고 앞으로 무엇이든 만들 수 있을 것 같은 자신감과 뜨거운 열정(전공에 대한 엄청난 자부심)을 얻었다. 또한 발표를 통해 많은 사람들 앞에서 직접 만든 결과물을 가지고 발표를 하는 것도 굉장히 큰 경험이다. 부족하고 완성도가 낮은 작품이라 해도 사람들 앞에서 발표할 때 자신이 만든 작품에 대해 자부심과 높은 자존감을 갖고 발표를 하는 것이 무엇보다도 중요하다는 것을 알았다(인생에서도 마찬가지 같다). 따라서 앞으로도 많은 발표를 통해 이러한 능력을 기르고 싶다. 또한 이번 프로젝트에서 만든 결과물의 훌륭함 보다는 팀원과 함께 협력하며 의견을 조율하고 회의를 거쳐서 같이 만들어 나갔다는 것이 가장 큰 의미가 있다. 우리는 대학에 들어와서 대학에서 배운 것들로 처음으로 팀원과 함께 결과물을 만들었기 때문이다. 그리고 무작정 개념공부만 하는 것보다 프로젝트를 진행하면서 배운 것들을 활용하여 프로젝트를 진행해보는 것이 실력향상에 큰 도움이 된다는 것도 알게 되었다. 제대로 된 결과물을 한번 만들고 보니 무언가를 공부해서 계속해서 만들어 보고 싶은 욕구가 솟구친다.

4 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	개념	App Inventor	국민대학교	From 2019.03.02	최진우	
2	개념	WSGI server	국민대학교	From 2019.05.15	최진우	

프로젝트 준비하는것과 발표하는 것 정말 재미있었습니다!!!