

소프트웨어 프로젝트 1 AD PROJECT 보고서

팀

우리 같이 A받자 흥흥흥~

조원

20191572 김지홍

20191572 김찬미

목차

- 게임 소개
- 게임 개발
 - *WSGI Script*
 - *App Inventor 블록코딩*
 - *단말기 연결*
- 제한요소 및 그 해결방안

팀 구성 및 역할 분담

20191572 김지홍

- App Inventor 블록코딩
- 보고서 작성

20191574 김찬미

- WSGI Script를 사용한 서버 구현
- 프로젝트 발표
- 보고서 작성

프로젝트명 및 게임 소개

" 찬미야 A받자 "

- 첫번째 게임: 달려라 찬미!
 - # 제한시간 안에 교실에 도착하게 만드는 게임
 - # 화면을 빠르게 터치해 캐릭터를 움직인다.
- 두번째 게임: 문제를 맞춰라 찬미!
 - # 풀고 싶은 문제를 선택해 문제를 보고 답을 맞추는 게임
 - # 문제를 선택하고 그 안에 객관식 문제를 푼다.

게임 개발

WSGI Script

```
from wsgiref.simple_server import make_server
from cgi import parse_qs, escape
import json
import pickle

def application(environ, start_response):
    1 total_coin = 0
    2 try:
        f = open("coin.txt", "rb")
        total_coin = pickle.load(f)
    3 except:
        f = open("coin.txt", "wb")
        pickle.dump(total_coin, f)
        f.close()
    4 else:
        f.close()
```

1)

total_coin 을 0으로 설정한다.

2)

파이썬 서버에 있는 coin.txt 파일을 이진모드로 읽는다. (rb 모드)

pickle.load()를 이용하여 coin.txt파일에 있는 내용을 total_coin 값에 넣는다.

3)

처음에 존재하지 않는 파일을 읽으려고 할 경우 Error가 발생한다.

만약 Error가 발생한 경우, coin.txt 파일을 이진모드로 입력한다. (wb모드)

pickle.dump(,)를 이용하여 coin.txt파일에 total_coin 값을 넣는다. 마무리로 파일을 close한다.

4)

예외 처리 코드(except)가 실행되지 않은 경우 수행하는 코드로서, 파일을 close한다.

게임 개발

WSGI Script

```
5 f = open("coin.txt", "rb")
  total_coin = pickle.load(f)
  f.close()

6 d = parse_qs(environ['QUERY_STRING'])
  xcoin = escape(d.get('coin', [''])[0])
  mode = escape(d.get('mode', [''])[0])
```

5)

이제는 파일이 '무조건' 존재하기에 coin.txt를 이진파일로 읽는다. (rb모드)

pickle.load()를 이용하여 coin.txt파일에 있는 내용을 total_coin값에 넣는다.
마무리로 파일을 close한다.

6)

Query string으로부터 두 인자를 받아들이고 해석한다.

게임 개발

WSGI Script

```
7 if(len(xcoin)>0):
    coin = int(xcoin)
8 else:
    coin = 0

9 if mode == "add":
    total_coin += coin
    elif mode == "minus":
        total_coin -= coin
        10 if(total_coin<0):
            total_coin = 0
11 elif mode == "init":
    total_coin = 0
```

7)

coin 값을 +,- 하지 않고 그대로 불러오기만 하는 경우,
Query뒤의 url은 'mode=get' 형태로 오며, coin은 값을 받
지 않는다.그러면 coin은 정수 형태의 값을 받지 않아, Error가 발
생한다.

따라서 xcoin의 길이가 1이상일 경우에만 coin값을 받아오도록 한
다.

8)

coin값을 받아오지 않을 때는 coin을 0으로 설정한다.

9)

mode가 'add' 일 경우에는 total_coin에 coin값을 더한다.

mode가 'minus' 일 경우에는 total_coin에 coin값을 뺀다.

10)

단, Coin의 값이 음수가 될 수 없으므로, total_coin이 0보다 작
으면 total_coin을 0으로 설정한다.

11)

게임 실행 중에는 쓰이지 않지만, mode가 'init' 일 경우
Coin의 값을 초기화할 수 있도록 설정한다.

게임 개발

WSGI Script

```
12 f = open("coin.txt", "wb")
    pickle.dump(total_coin, f)
    f.close()

13 response_body = json.dumps({'coin': total_coin, "mode": mode})
    status = '200 OK'
    response_headers = [
        ('Content_Type', 'application/json'),
        ('Content_Length', str(len(response_body)))
    ]
    start_response(status, response_headers)
    return [response_body]
```

```
httpd = make_server('192.168.0.5', 8051, application)
httpd.serve_forever()
```

12)

total_coin에 coin값을 더하거나 뺐기에, 그 값을 파일에 저장하기 위해 coin.txt파일을 이진모드로 입력한다. (wb모드)

pickle.dump(,)를 이용하여 coin.txt파일에 total_coin 내용을 넣는다.

마무리로 파일을 close한다.

13)

HTTP 응답을 위해 json 문자열을 만들고 헤더 정보를 설정한다.

실행 및 확인

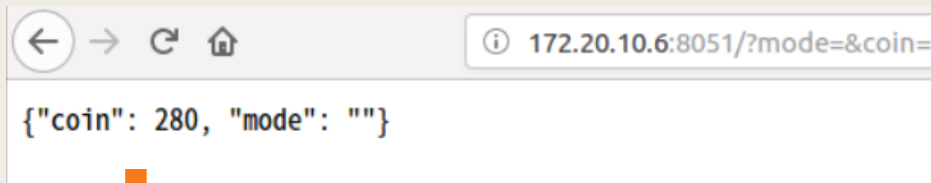
```
chanmi@chanmi-ThinkPad-T480s:~/Projects/py/myflaskapp$ python totalcoin.py
```

WSGI 스크립트를 실행시킨다.

커서가 깜박이면 실행되고 있다는 뜻이다.



브라우저 주소 창에 " **IP주소**:8051/?mode= &coin= " 형태로 URL을 입력한다.



실행 완료



브라우저 주소 창의 coin은 +, - 하는 'coin' 값을, 출력된 창의 coin은 'total_coin' 을 의미한다.

HTTP응답을 받을 때 json.dump({ 'coin' : total_coin})을 했기 때문이다.

파일을 만들어서 입력하고 저장한 파일의 내용을 불러오는 것이기에, 연결이 끊겼다가 다시 접속해도 값이 유효하다는 점에서 애플리케이션의 tiny DB와 유사한 역할을 한다고 볼 수 있다.

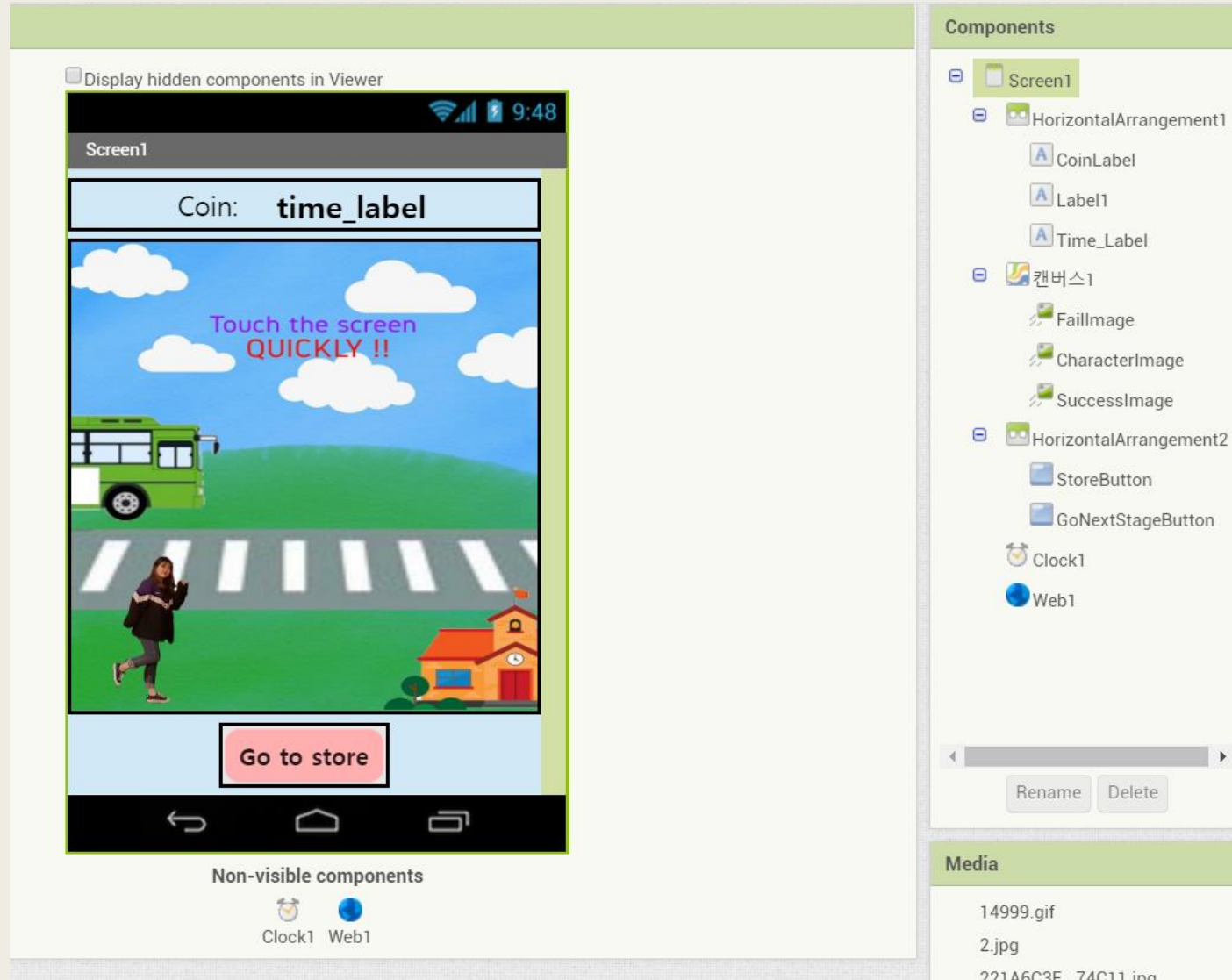
앱 디자인 - Screen 1

스크린이 바뀌어도 값이 초기화되지 않는 코인Label을 위에 만들어준다.

언제든지 매점에 갈 수 있는 매점버튼을 추가해준다.

제한시간을 보여주는 time_Label을 위에 위치해두고 시계기능을 추가한다.

캔버스에 배경사진과 캐릭터 사진을 넣어준다.



블록코딩 할 때 혼란을 줄이기 위하여 각 Label, Button, Textbox의 이름들을 바꿔준다.

서버 URL로부터 코인값을 받아오기 위하여 Web 기능을 추가해준다.

앱이 시작했을 때 당장 보이지 않는 이미지들은 non-visible로 설정해준다.

블록코딩 - Screen 1

initialize global url to `http://172.20.10.6:8051/?`

initialize global coin to `0`

initialize global time to `10`

when Screen1.Initialize

do
 set Web1.Url to `join` `get global url`
 `mode=get`
 call Web1.Get

when 캔버스1.Touched

`x` `y` touchedAnySprite
do
 set CharacterImage.X to `CharacterImage.X` + `6`

반복해서 이용하는 url을 변수 설정한다.

처음 global coin의 값을 0으로 설정한다.

처음에 global time의 값을 10으로 설정해 준다.

Screen이 시작될 때 url을 요청하여 저장된 Coin값을 불러 올 수 있도록 한다.

캔버스가 터치됐을 때 Character Image가 x좌표 +6이 되도록 한다.

블록코딩 - Screen 1



처음에 10으로 설정해둔 global time이 1씩 줄어들도록 한다.

Time_Label에 global time값을 보인다.

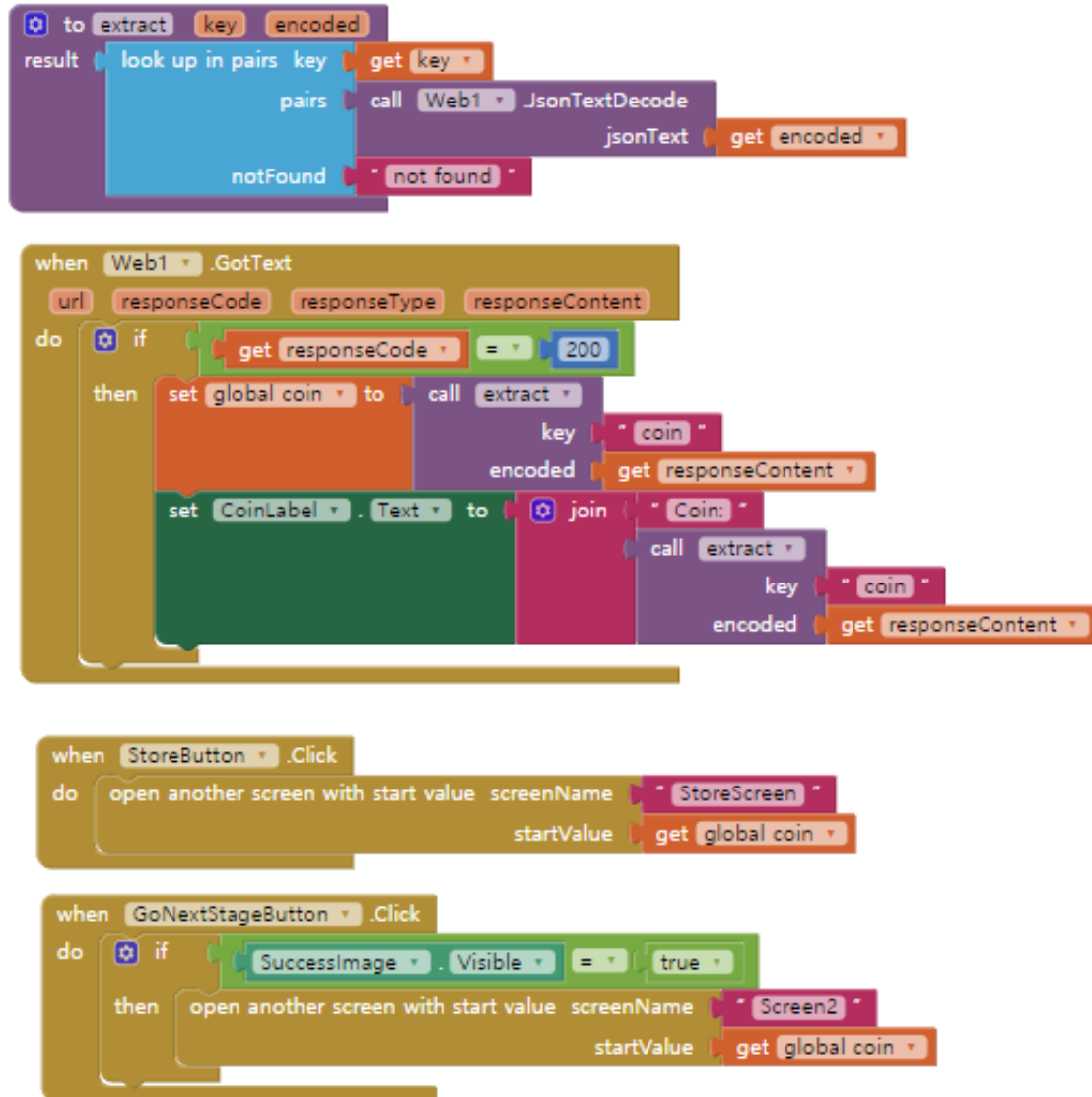
Global time이 0이 되는 순간, 시계 컴포넌트를 멈추며 Fail image를 보여주고 global time이 음수 값이 되지 않도록 한다.

Non-visible로 설정 되어 있던 Fail Image를 visible로 바꾼다.

visible로 설정 되어 있던 Character Image를 non-visible로 바꾼다.

게임의 결과를 나타내는 이미지가 visible되면 GoNextStage 버튼이 나타나게끔 한다.

블록코딩 – Screen 1



Json 디코딩, 값 추출하는 프로시저를 만든다.

url에 연결할 때, reponseCode가 200이면

global coin의 값을 key가 'coin' 인 값의 responseContent로 설정한다.

CoinLabel을 'Coin:' 과 key가 'coin' 인 값의 responseContent를 연결하여 설정한다.

위에서 설명하는 두 블록은 json값이 필요한 모든 스크린에 추가해서 각 스크린마다 값을 받아올 수 있게끔 해준다.

StoreButton이 눌렸을 때, StoreScreen으로 이동하며 globalcoin의 값을 받아간다.

GoNextButton이 눌렸을 때,
SuccessImage가 visible이라면,
Screen2로 이동하며 globalcoin의 값을 받아간다.

블록코딩 - Screen1

Screen1의 전체
블록코딩 모습이다.

initialize global url to `"http://172.20.10.6:8051/"`

initialize global coin to `0`

initialize global time to `10`

when Screen1.Initialize
do set Web1.Url to `join` `get global url` `"mode=get"`
call Web1.Get

when 캔버스1.Touched
do set CharacterImage.X to `CharacterImage.X + 6`

when Clock1.Timer
do set global time to `get global time - 1`
set Time_Label.Text to `get global time`
if `get global time <= 0`
then set Clock1.TimerEnabled to `false`
set FailImage.Visible to `true`
set CharacterImage.Visible to `false`
set GoNextStageButton.Visible to `true`

when CharacterImage.EdgeReached
edge
do if `get global time > 0`
then set SuccessImage.Visible to `true`
set Web1.Url to `join` `get global url` `"mode=add&coin=30"`
call Web1.Get
else set Web1.Url to `join` `get global url` `"mode=get"`
call Web1.Get
set CharacterImage.Visible to `false`
set GoNextStageButton.Visible to `true`

to extract key encoded
result look up in pairs key `get key`
pairs call Web1.JsonTextDecode
jsonText `get encoded`
notFound `"not found"`

when Web1.GotText
url responseCode responseType responseContent
do if `get responseCode == 200`
then set global coin to `call extract` `key "coin"` `encoded get responseContent`
set CoinLabel.Text to `join` `"Coin:"` `call extract` `key "coin"` `encoded get responseContent`

when StoreButton.Click
do open another screen with start value screenName `"StoreScreen"`
startValue `get global coin`

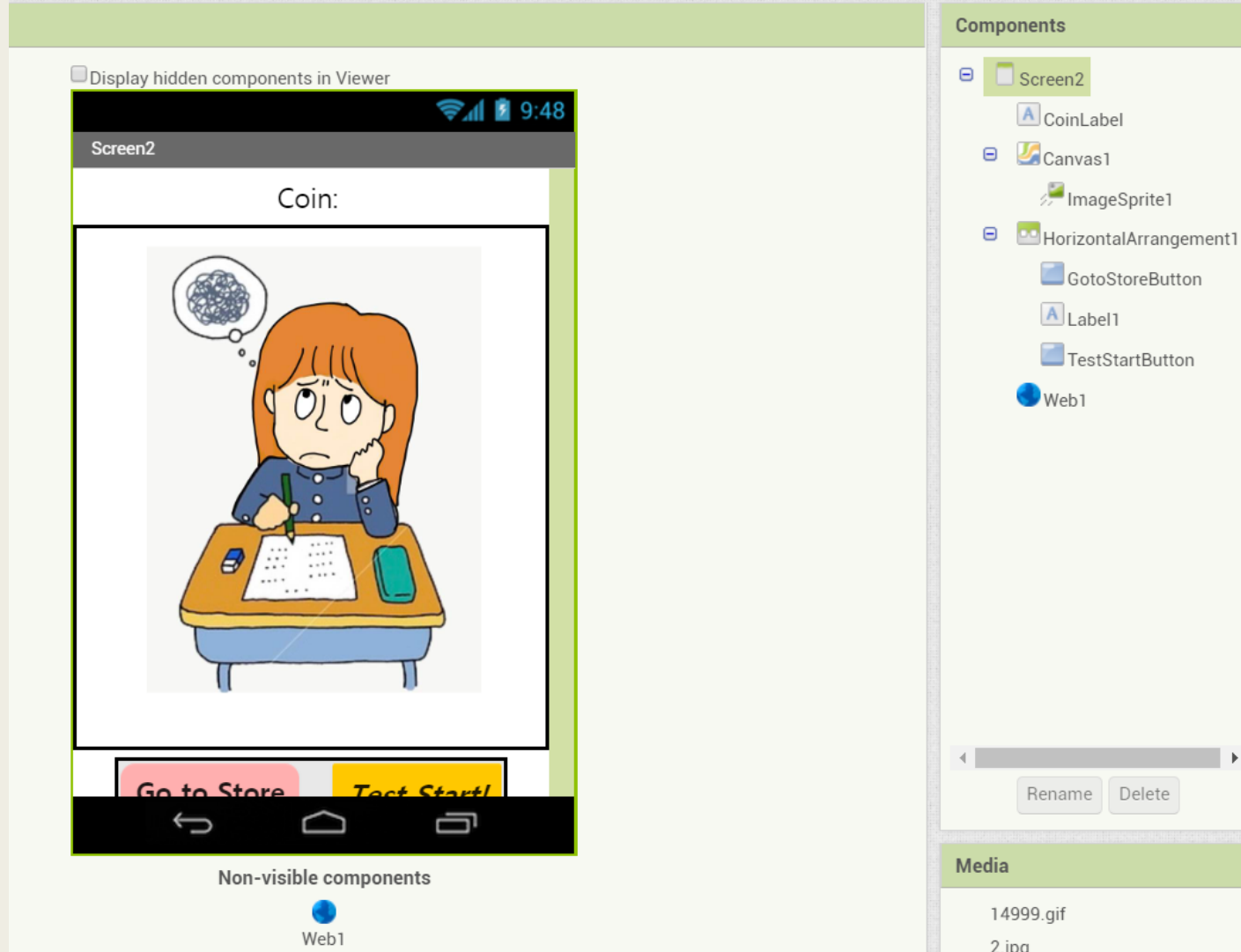
when GoNextStageButton.Click
do if `SuccessImage.Visible == true`
then open another screen with start value screenName `"Screen2"`
startValue `get global coin`

앱 디자인 - Screen2

스크린이 바뀌어도 값이 초기화되지 않는 코인Label을 위에 만들어준다.

언제든지 매점에 갈 수 있는 매점버튼을 추가해준다.

캔버스에 배경사진과 캐릭터 사진을 넣어준다.

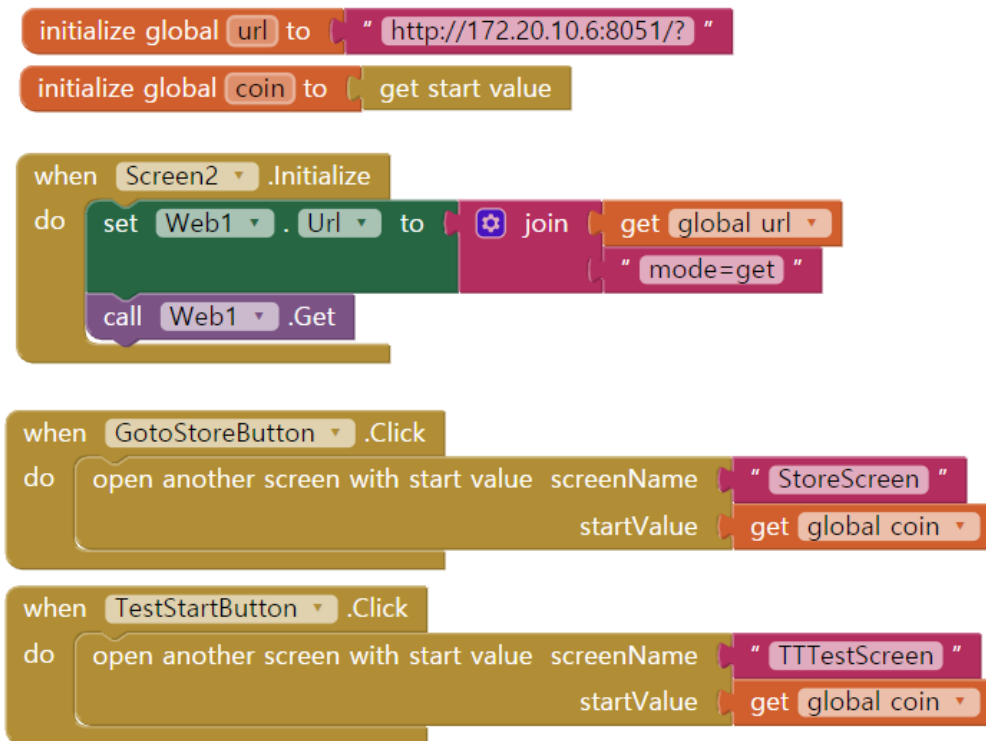


블록코딩 할 때 혼란을 줄이기 위하여 각 Label, Button, Textbox의 이름들을 바꿔준다.

서버 URL로부터 코인값을 받아오기 위하여 Web 기능을 추가해준다.

문제 푸는 화면으로 넘어갈 수 있게 해주는 버튼을 만들어준다.

블록코딩 - Screen2



The image shows a Scratch script for a screen named 'Screen2'. It starts with two global variables, 'url' and 'coin', being initialized. 'url' is set to 'http://172.20.10.6:8051/?' and 'coin' is set to 'get start value'. Then, a 'when Screen2.Initialize' event triggers a 'do' block containing three steps: 'set Web1.Url to' followed by a 'join' block that concatenates 'get global url', 'mode=get', and a quote. Finally, 'call Web1.Get' is executed. Below this, a 'when GotoStoreButton.Click' event triggers a 'do' block that opens 'StoreScreen' with 'start value' set to 'get global coin'. A third event, 'when TestStartButton.Click', triggers a 'do' block that opens 'TTTestScreen' with 'start value' set to 'get global coin'.

```
initialize global url to "http://172.20.10.6:8051/?"  
initialize global coin to get start value  
  
when Screen2.Initialize  
do  
  set Web1.Url to join [get global url] [mode=get]  
  call Web1.Get  
  
when GotoStoreButton.Click  
do  
  open another screen with start value screenName "StoreScreen"  
  start value get global coin  
  
when TestStartButton.Click  
do  
  open another screen with start value screenName "TTTestScreen"  
  start value get global coin
```

반복해서 이용하는 url을 변수 설정한다.

global coin의 값을 받아온다.

Screen2가 실행 됐을 때,

지정된 global url에 특정 mode와 coin값을
설정하지 않고 url을 완성한다.

Web을 이용하여 url에 연결한다.

StoreButton이 눌렸을 때, StoreScreen으로
이동하며 globalcoin의 값을 받아간다.

TestStartButton이 눌렸을 때,
TTTestScreen으로 이동하며 globalcoin의 값을
받아간다.

앱 디자인 - Store Screen





스크린이 바뀌어도
값이 초기화되지
않는 코인Label을
위에 만들어준다.

캔버스에
배경사진과 매점음식
사진들을 추가해준다.

☐ Display hidden components in Viewer

StoreScreen

Coin:
Choose your menu

 마카롱 coin:20	 커피 coin:10
 연바 coin:20	 스무디 coin:15

Non-visible components

Components

StoreScreen

CoinLabel

ExplainLabel

TableArrangement1

- VerticalArrangement1
 - 마카롱Canvas
 - 마카롱Button

VerticalArrangement2

- 커피Canvas
- 커피Button

VerticalArrangement3

- 연바Canvas
- 연바Button

VerticalArrangement4

- 스무디Canvas
- 스무디Button

Web1

Rename

Delete

Media

14999.gif

블록코딩 할 때 혼란을
줄이기 위하여 각
Label, Button,
Textbox의 이름들을
바꿔준다.

서버 URL로부터
코인값을 받아오기
위하여 Web 기능을
추가해준다.

블록코딩 - Store Screen

```
initialize global url to "http://172.20.10.6:8051/?"  
initialize global coin to get start value  
  
when StoreScreen.Initialize  
do  
  set Web1.Url to join [get global url] ["mode=get"]  
  call Web1.Get  
  
when 마카롱Button.Click  
do  
  set Web1.Url to join [get global url] ["mode=minus&coin=20"]  
  call Web1.Get  
  
when 커피Button.Click  
do  
  set Web1.Url to join [get global url] ["mode=minus&coin=10"]  
  call Web1.Get  
  
when 연바Button.Click  
do  
  set Web1.Url to join [get global url] ["mode=minus&coin=20"]  
  call Web1.Get  
  
when 스무디Button.Click  
do  
  set Web1.Url to join [get global url] ["mode=minus&coin=15"]  
  call Web1.Get
```

반복해서 이용하는 url을 변수 설정한다.
global coin의 값을 받아온다.

StoreScreen이 실행 됐을 때,
지정된 global url에 특정 mode와 coin값을
설정하지 않고 url을 완성한다.
Web을 이용하여 url에 연결한다.

각 마카롱, 커피, 연바, 스무디의 버튼을 누를 때,
지정된 global url에 minus mode와 coin값을
설정하여 url을 완성한다.
Web을 이용하여 url에 연결한다.

블록코딩 - Store Screen

Store Screen의 전체 블록코딩 모습이다.

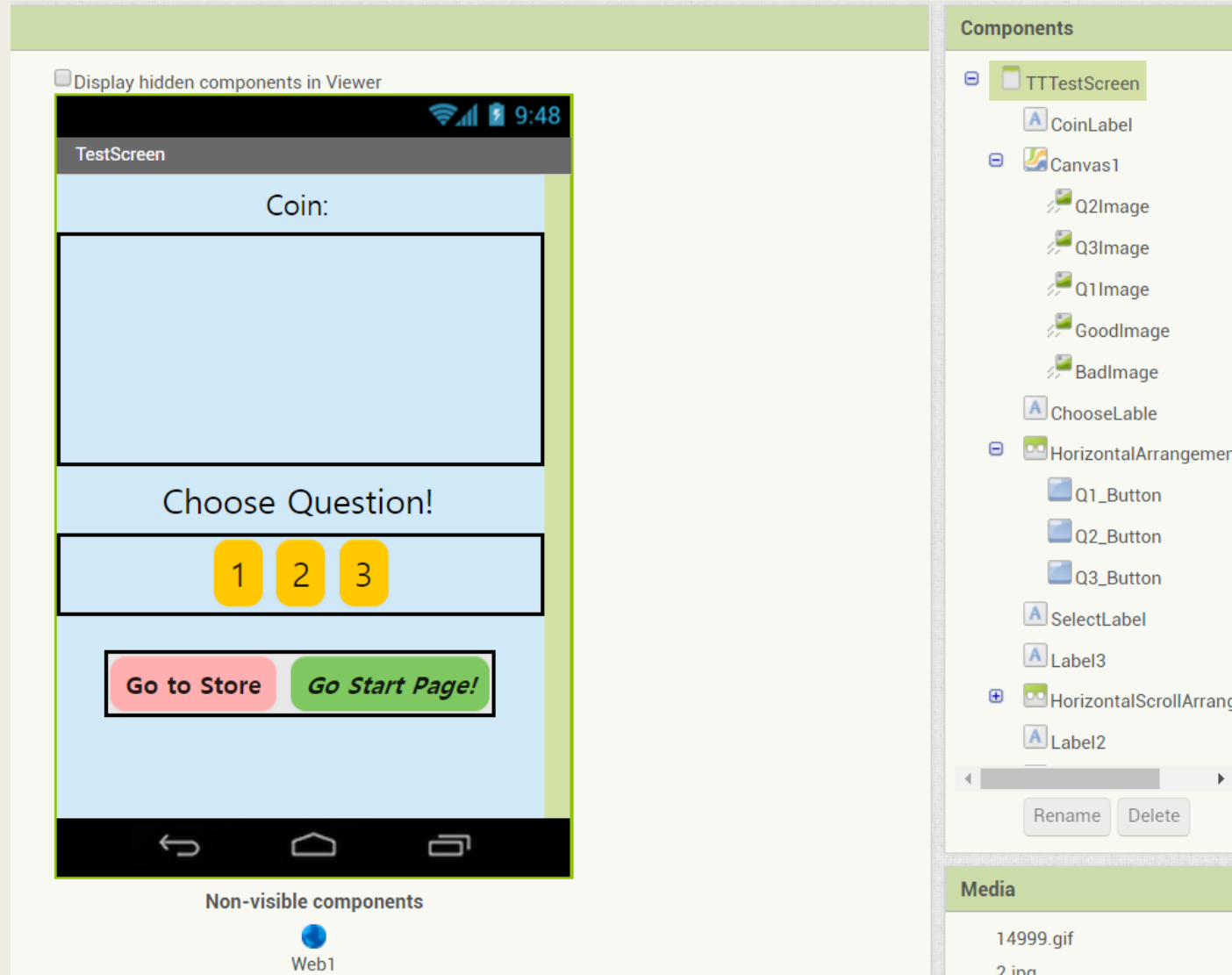
The image displays a collection of Scratch-style block code for a 'Store Screen' application. The code is organized into several sections:

- Initialization:**
 - `initialize global url to "http://172.20.10.6:8051/?"`
 - `initialize global coin to get start value`
- When StoreScreen.Initialize:**
 - do: `set Web1.Url to join [get global url] [mode=get]`
 - `call Web1.Get`
- When 마카롱Button.Click:**
 - do: `set Web1.Url to join [get global url] [mode=minus&coin=20]`
 - `call Web1.Get`
- When 커피Button.Click:**
 - do: `set Web1.Url to join [get global url] [mode=minus&coin=10]`
 - `call Web1.Get`
- When 연바Button.Click:**
 - do: `set Web1.Url to join [get global url] [mode=minus&coin=20]`
 - `call Web1.Get`
- When 스무디Button.Click:**
 - do: `set Web1.Url to join [get global url] [mode=minus&coin=15]`
 - `call Web1.Get`
- Web1.GetText:**
 - url, responseCode, responseType, responseContent
 - do: `if [get responseCode] [200]`
 - then: `set CoinLabel.Text to join ["Coin: "] [call procedure]`
 - call procedure: `key "coin", encoded [get responseContent]`
 - `set global coin to call procedure`
 - call procedure: `key "coin", encoded [get responseContent]`
- Procedure: key encoded:**
 - result: `look up in pairs key [get key], pairs [call Web1.JsonTextDecode jsonText [get encoded]]`
 - `notFound "not found"`

앱 디자인 - Test Screen

스크린이 바뀌어도
값이 초기화되지
않는 코인Label을
위에 만들어준다.

캔버스에
문제번호와 문제를 풀
수 있는 버튼들을
추가해준다.



블록코딩 할 때 혼란을
줄이기 위하여 각
Label, Button,
Textbox의 이름들을
바꿔준다.

서버 URL로부터
코인값을 받아오기
위하여 Web 기능을
추가해준다.

블록코딩 – Test Screen

```
when Q1_Button .Click
do
  set Canvas1 . Height to 300
  set ChooseLabel . Visible to false
  set HorizontalArrangement1 . Visible to false
  set Q1Image . Visible to true
  set SelectLabel . Visible to true
  set HorizontalScrollArrangement1 . Visible to true
```

```
when Q2_Button .Click
do
  set Canvas1 . Height to 300
  set ChooseLabel . Visible to false
  set HorizontalArrangement1 . Visible to false
  set Q2Image . Visible to true
  set SelectLabel . Visible to true
  set HorizontalScrollArrangement1 . Visible to true
```

```
when Q3_Button .Click
do
  set Canvas1 . Height to 300
  set ChooseLabel . Visible to false
  set HorizontalArrangement1 . Visible to false
  set Q3Image . Visible to true
  set SelectLabel . Visible to true
  set HorizontalScrollArrangement1 . Visible to true
```

각각의 문제가 눌렀을 때, 초기에 height가 200이었던 canvas의 height를 300으로 변경한다.

문제를 선택하라는 ChooseLabel과 문제를 선택하는 arrangement가 non-visible된다.

선택된 문제의 이미지가 visible된다.

답을 고르라는 SelectLabel과

답을 고를 수 있는 arrangement가 visible된다.

블록코딩 – Test Screen



The image shows two Scratch code blocks. The first block is a 'when clicked' block for 'GotoStoreButton'. It contains a 'do' block with 'open another screen with start value'. The 'screenName' field is set to 'StoreScreen' and the 'startValue' field is set to 'get global coin'. The second block is a 'when clicked' block for 'GoStartPageButton'. It contains a 'do' block with 'open another screen with start value'. The 'screenName' field is set to 'Screen2' and the 'startValue' field is set to 'get global coin'.

```
when GotoStoreButton .Click
do
  open another screen with start value
    screenName "StoreScreen"
    startValue get global coin

when GoStartPageButton .Click
do
  open another screen with start value
    screenName "Screen2"
    startValue get global coin
```

- # StoreButton이 눌렸을 때, StoreScreen으로 이동하며 globalcoin의 값을 받아간다.
- # GotoStartButton이 눌렸을 때, Screen2로 넘어가 다른 문제를 풀 수 있도록 해주며 globalcoin의 값을 받아간다.

블록코딩 - Test Screen

```
when a_Button.TouchDown
do
  if Q1Image.Visible = true
  then
    set GoodImage.Visible to true
    set Web1.Url to join get global url "mode=add&coin=20"
    call Web1.Get
  else
    set BadImage.Visible to true
  end if
  open another screen screenName "Screen2"
```

```
when b_Button.TouchDown
do
  if Q3Image.Visible = true
  then
    set GoodImage.Visible to true
    set Web1.Url to join get global url "mode=add&coin=20"
    call Web1.Get
  else
    set BadImage.Visible to true
  end if
  open another screen screenName "Screen2"
```

```
when c_Button.TouchDown
do
  if Q2Image.Visible = true
  then
    set GoodImage.Visible to true
    set Web1.Url to join get global url "mode=add&coin=20"
    call Web1.Get
  else
    set BadImage.Visible to true
  end if
  open another screen screenName "Screen2"
```

```
when d_Button.TouchDown
do
  set BadImage.Visible to true
  open another screen screenName "Screen2"
```

Q1의 문제 답 - a

Q2의 문제 답 - c

Q3의 문제 답 - b

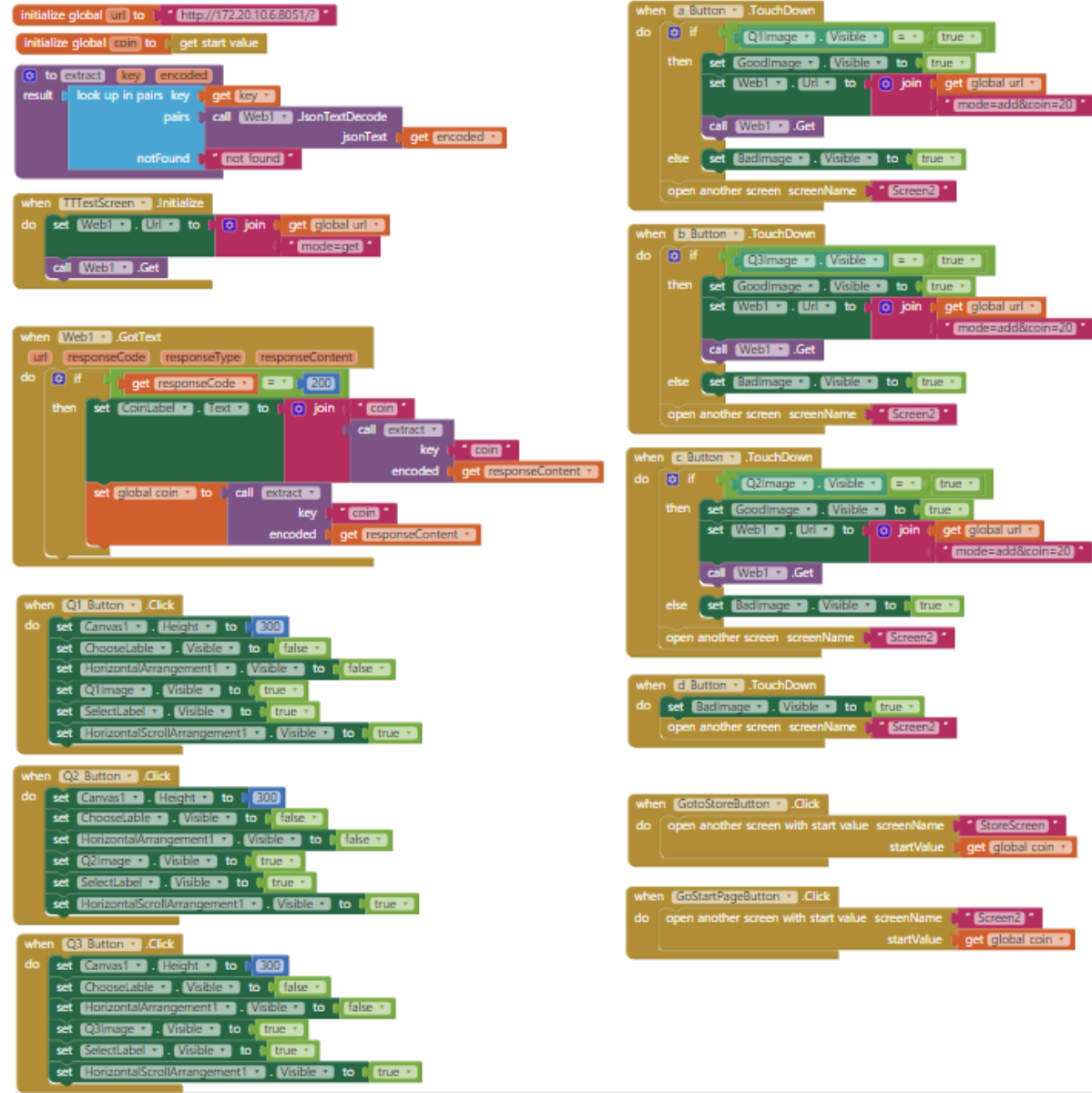
답 버튼을 눌렀을 때

1. 정답의 문제 이미지가 visible 되어 있다면,
Non-visible로 설정 되어 있던 GoodImage를 visible한다.
지정된 global url에 add mode와 coin값을 설정하여 url을 완성한다.
Web을 이용하여 url에 연결한다.

2. 아니라면
Non-visible로 설정 되어 있던 BadImage를 visible한다.
문제를 풀면 GoStartPage버튼을 누르지 않아도, Screen2로 이동한다.

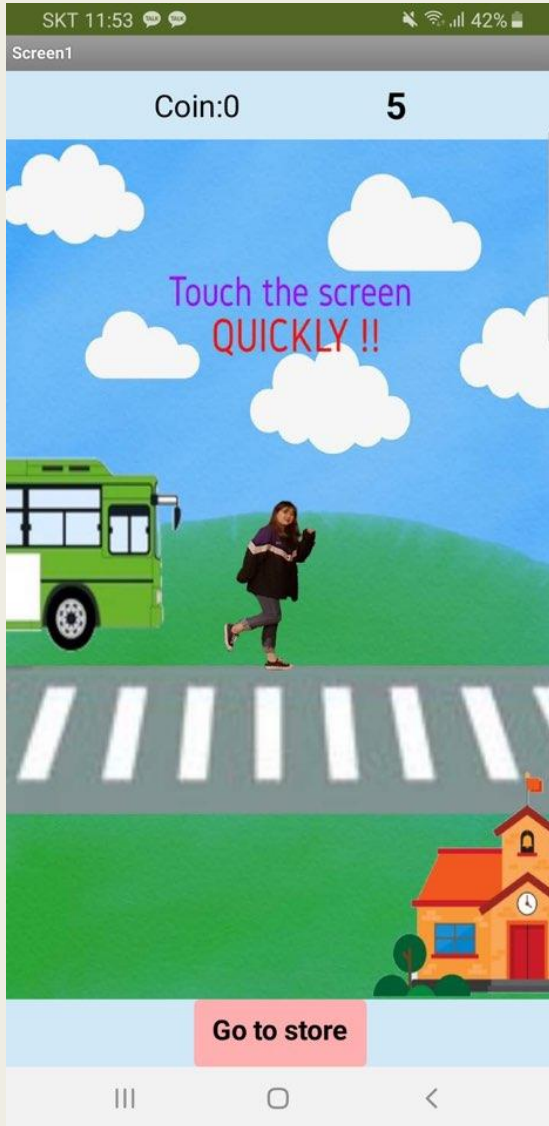
블록코딩 - Test Screen

Test Screen의
전체 블록코딩 모습이다.

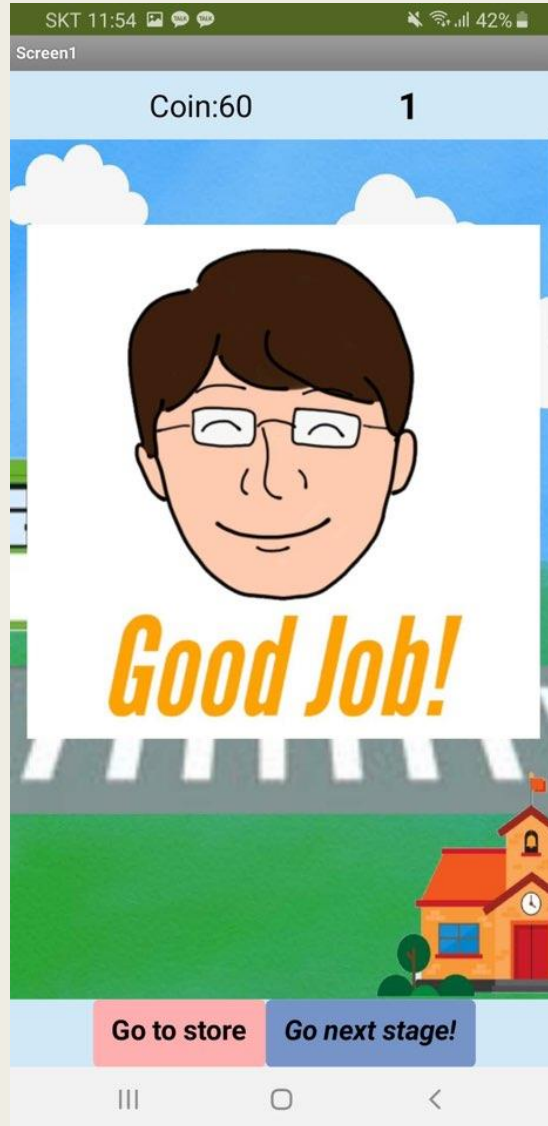


단말기연결 - Screen 1

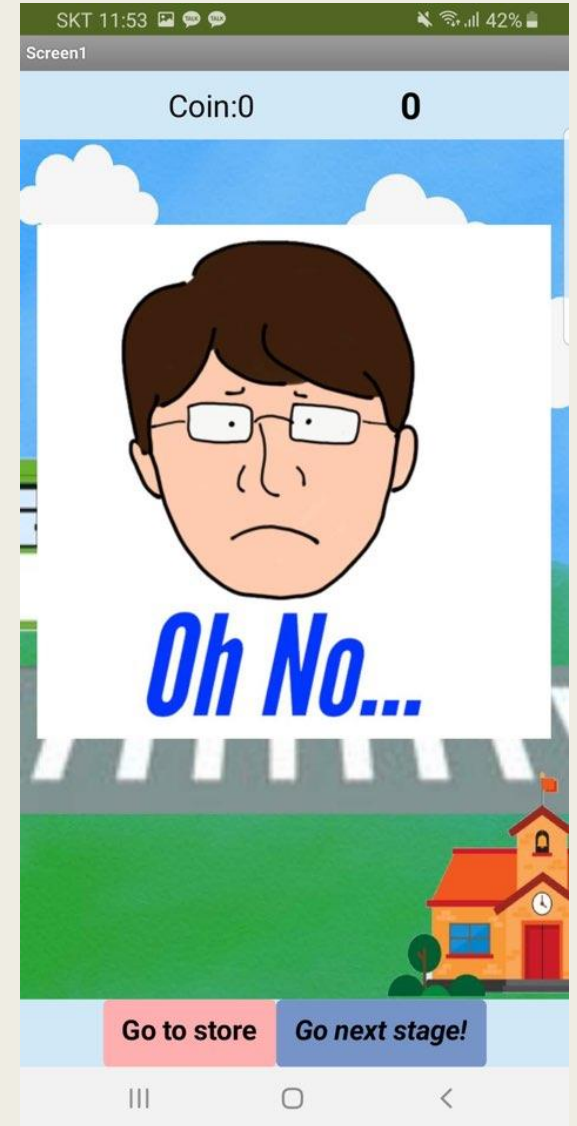
화면첫모습



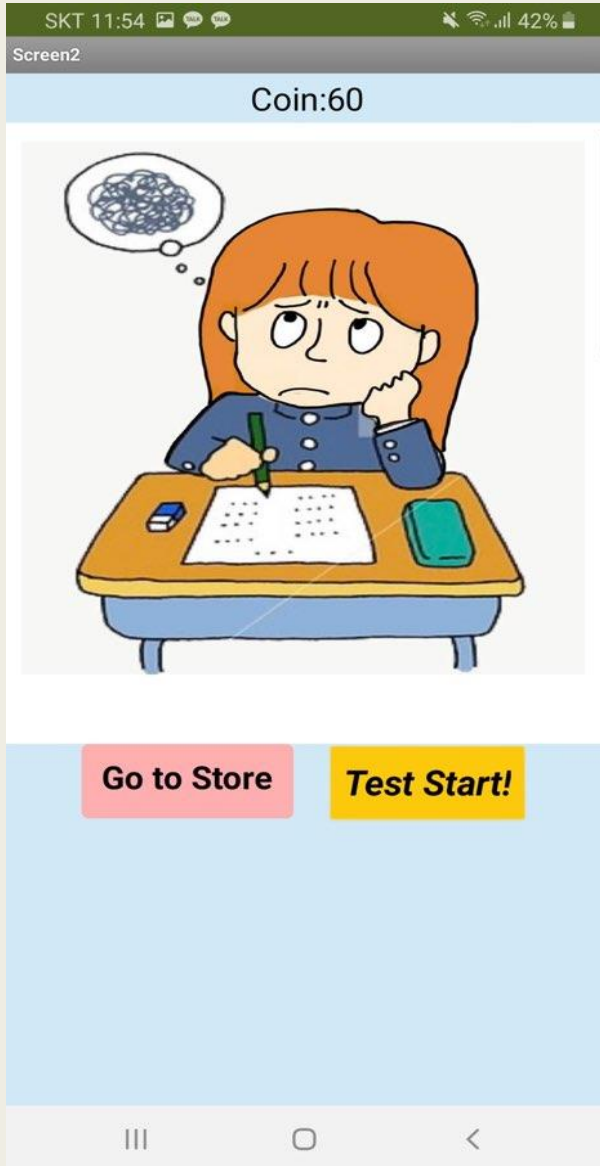
성공했을 때



실패했을 때



단말기연결 - Screen 2



Screen2의 화면은 이것뿐이다.

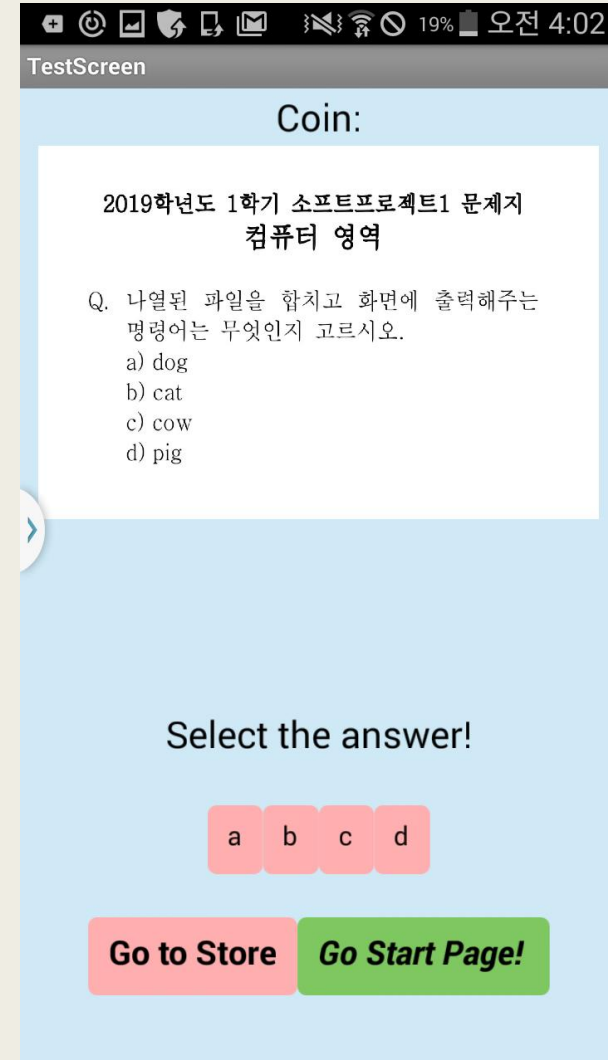
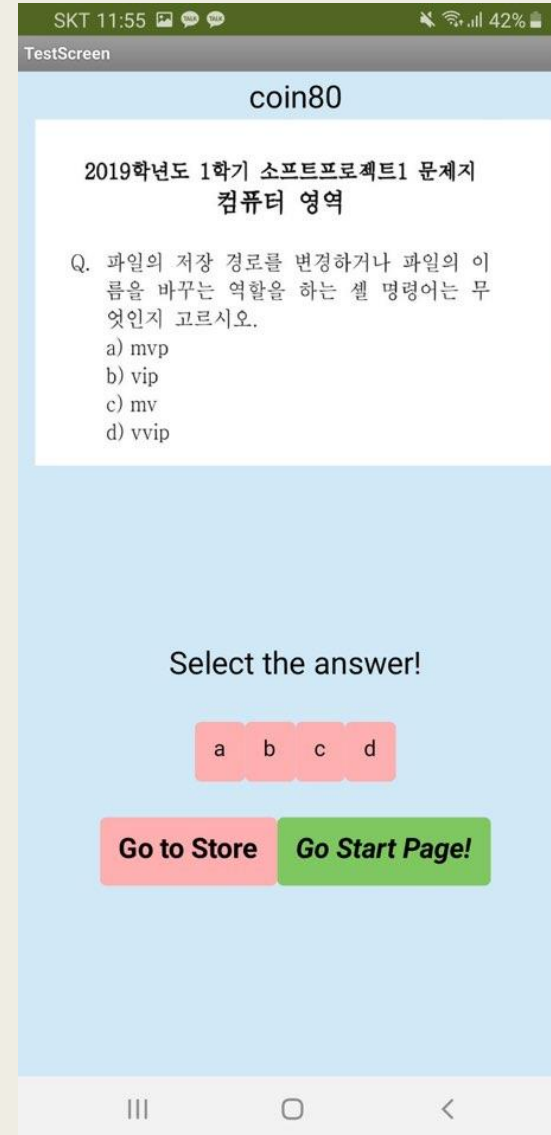
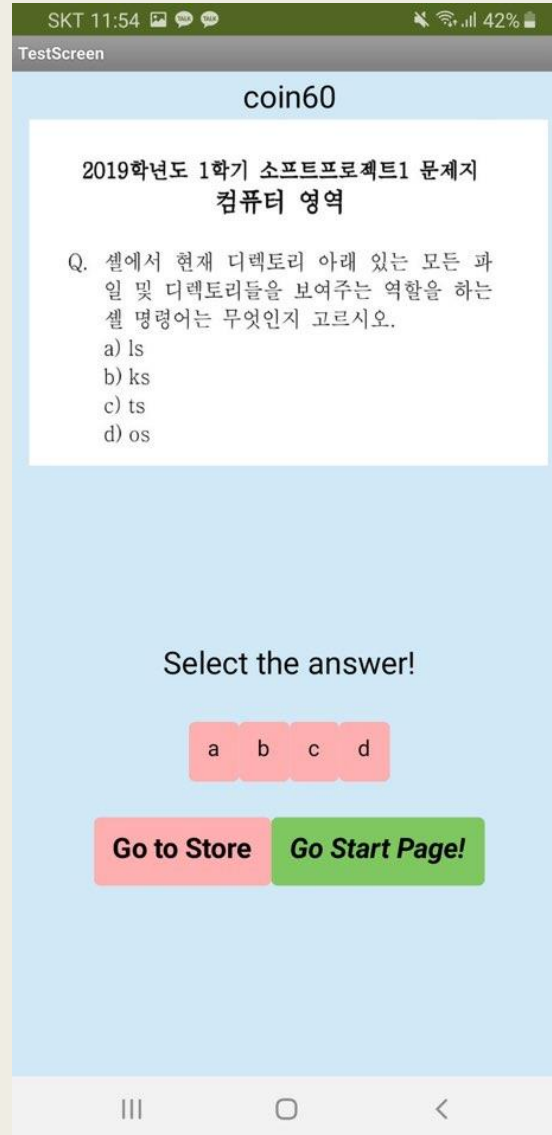
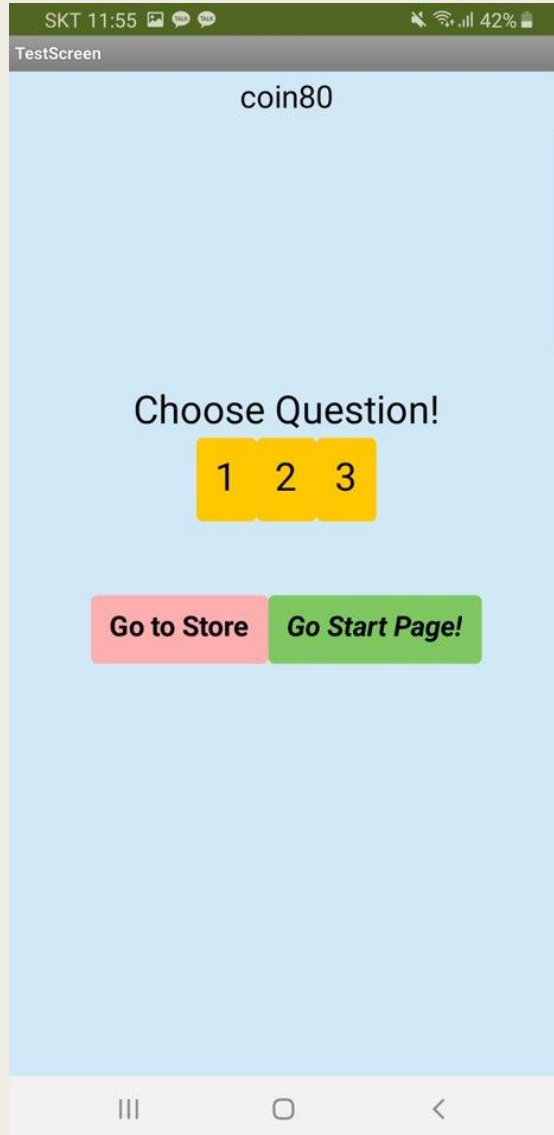
위에는 저장된 코인값이 뜬다.

매점에 갈 수 있는 버튼과 문제를 풀 수 있는 버튼이 두가지 존재한다.

버튼을 선택해 원하는 스크린으로 갈 수 있다.

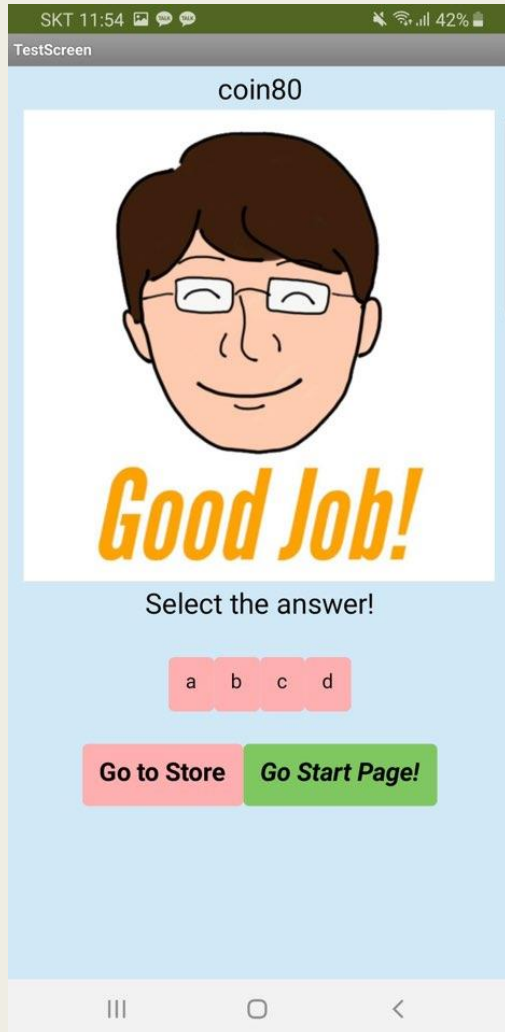
단말기연결 - Test Screen

testscreen의 문제들이다

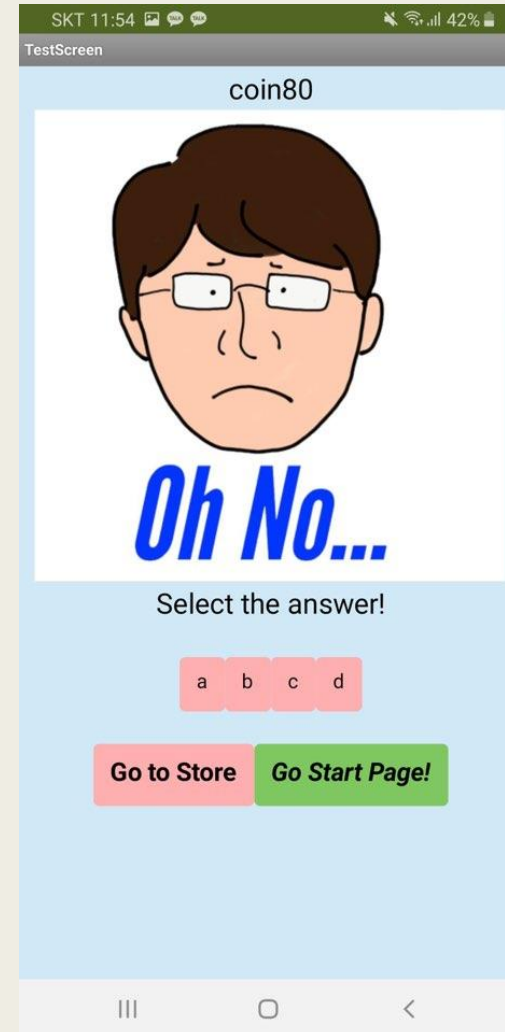


단말기연결 - Test Screen

문제를 맞췄을 때 화면



문제를 틀렸을 때 화면



단말기연결 - Store Screen



매점에 들어갔을 때의 화면이다.

위에는 저장된 코인값이 뜬다.

매점에서 음식을 사고싶을 땐 코인을 사용한다.

음식의 버튼을 누르면 코인의 값이 줄어드는 것을 옆의 사진 1,2를 통해 알 수 있다.

제한 요소 및 그 해결방안

1. Url에 mode와 coin의 값을 지정할 때 mode=minus일 경우, coin의 값이 줄어들었다.

그런데, 계속 minus를 하면 coin의 값이 음수의 형태로 출력될 수 있다.

서버에서 total_coin이 0보다 작을 경우, total_coin의 값을 0으로 표시했다.

2. 실제 게임 중에는 mode가 add이거나 minus인 경우가 쓰이지만,

Mode가 init일 경우를 설정하여 total_coin = 0이 되도록 설정한다.

이것은 파일에 직접 들어가지 않아도, total_coin을 초기화 할 수 있어 개발자 입장에 편리하다.