

소프트웨어 프로젝트 1

학기말 프로젝트

20181570 1분반 강주희

1. 요구 사항 명세

- 기능적 요구사항

1. 비밀번호에 대한 최대값(경계)을 사용자가 설정할 수 있어야 함.
2. 숫자를 맞추고 있는 중이 아닐 때는 버튼을 비활성화 시켜야 함.
3. 새로운 게임을 시작하는 버튼과
숫자를 맞추는 버튼이 따로 존재 해야함.
4. 정답을 맞추면 새로운 게임이 시작하기 전의 상태로 돌아가야함

서버 API 설계

1. 새로운 게임 시작

URL : <http://172.20.10.3/game/new>

Request Method : HTTP POST

입력인자

boundary : 맘 속의 숫자의 최대 값

출력형태

성공하는 경우

```
{"code": "success"}
```

실패하는 경우

```
{"code": "error" , "msg": MSG}
```

MSG : 실패했을 때 출력되는 문자열

-0, 음수, 문자열이 입력됐을때

“유효한 범위가 아닙니다.”

서버 API 설계

2. 사용자에게 의한 맞추기 시도

URL : <http://172.20.10.3/game/guess>

Request Method : HTTP POST

입력인자

guess : 사용자의 의한 맞추기 시도 (숫자)

출력형태 (json)

성공하는 경우

```
{"msg":"number Answer : smaller or greater",  
  "code":success}
```

실패하는 경우

```
{"msg:" : MSG,  
  "code" : error }
```

! MSG

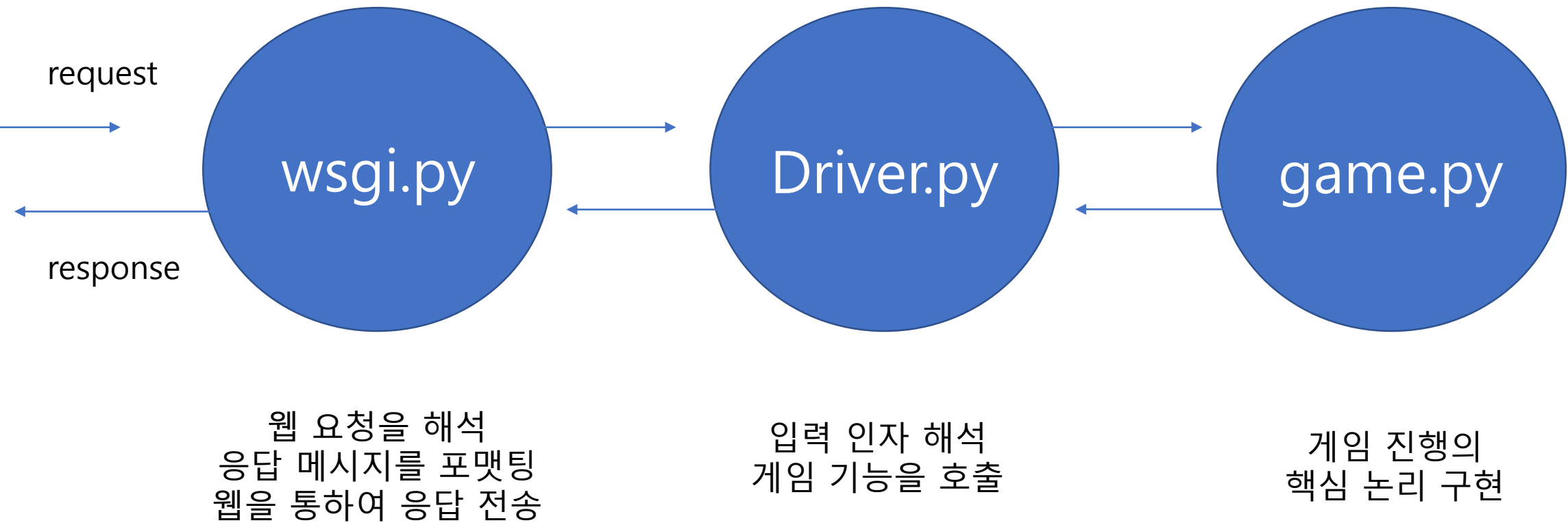
-0 또는 boundary를 초과한 값을 입력한 경우

You must enter a number over 0 and below boundary

-문자열을 입력한 경우

유효한 값이 아닙니다.

서버 SW 구조 설계



게임의 핵심 로직 설계

- class My_Secret_Number:

이 클래스가 가져야 할 속성들은 ?

self.__trial # 현재까지 시도 횟수

self.__secret # 비밀 숫자

self.__boundary #비밀 숫자의 최댓값

이 클래스가 가져야할 메서드들은 ?

def __init__(self): # 객체의 인스턴스 생성시, 알려진 초기 상태에 있게 함

def New_Game(self,boundary): # 새로운 게임을 시작

def Guess(self,userguess): # 맞추기 시도

def Get_Trials(self): #현재까지 시도 횟수 반환

게임 클래스의 뼈대 모습

- ```
class My_Secret_Number:
 def __init__(self):
 ...
 def New_Game (self,boundary):
 ...
 def Guess(self,userguess):
 ...
 def Get_Trials(self):
 ...
```

# 메서드의 구현 : `__init__`

- `class My_Secret_Number:`

```
def __init__(self):
```

```
 self.__trials = 0
```

```
 self.__secret = 0
```



# 메서드의 구현 : New\_Game()

- 메서드의 명세 설계 :

입력 : boundary – 맘 속의 숫자의 최댓값

출력 : X

해야할 일 들:

1. 1부터 boundary 까지의 범위에서 난수 생성
2. 시도 횟수는 0으로 초기화 한다.

# 메서드의 구현 : New\_Game()

```
Import random
```

```
class My_Secret_Number:
```

```
 def New_Game(self, boundary):
 self.__boundary = boundary
 self.__secret = random.randint(1,boundary)
 self.__trials = 0
```

# 메서드의 구현 : Guess()

- 메서드의 명세 설계 :

입력 : `userguess` – 사용자가 입력한 수

반환 : `userguess - self._secret`

`userguess - self._secret > 0` : "smaller"

`userguess - self._secret < 0` : "greater"

해야할 일 들:

1. 시도 횟수를 1만큼 증가시킨다.
2. 사용자가 입력한 수와 비밀숫자에 대한 연산을 반환한다.
3. 사용자가 입력한 수가 오류를 발생시킬 때 문구를 반환한다.

# 메서드의 구현 : New\_Game()

```
class My_Secret_Number:
```

```
 ...
```

```
 def Guess(self, userguess):
 self.__trials += 1
```

```
 userguess = int(userguess)
```

```
 if userguess > self.__boundary or userguess == 0 :
 return "You must enter a number over 0 and below {}".format(boundary)
```

```
 return userguess - self.__secret
```

# 메서드의 구현 : Get\_Trials()

지금까지 맞추기를 시도한 횟수를 반환

```
class My_Secret_Number:
```

```
...
```

```
def Get_Trials(self):
 return self.__trials
```

# 게임의 핵심 로직 구현, 테스트

```
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ python3 game.py
your guess : 50
smaller
your guess : 25
greater
your guess : 40
smaller
your guess : 30
greater
your guess : 35
SUCCESS in 5 tries
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
```

```
if __name__ == '__main__':
 s = My_Secret_Number()
 a = s.New_Game(100) # boundary = 100

 while True:
 input_number = input("your guess: ")
 result = s.Guess(input_number)
 if result == 0:
 break
 elif result < 0 :
 print("greater")
 elif result > 0 :
 print("smaller")
 else:
 print(result)
 print("SUCCESS in {} trials".format(s.Get_Trials()))
```

# WSGI 스크립트 및 추가 모듈 구현

```
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ curl "http://127.0.0.1/game/new" -d "boundary=100"
{"code": "success"}jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ curl "http://127.0.0.1/game/guess" -d "guess=0"
{"msg": "You must enter a number over 0 and below 100.", "code": "error"}jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ curl "http://127.0.0.1/game/guess" -d "guess=-20"
{"msg": "You must enter a number over 0 and below 100.", "code": "error"}jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ curl "http://127.0.0.1/game/guess" -d "guess=ddd"
{"msg": "\uc720\uud6a8\uud55c \uac12\uc774 \uc544\uub2d9\uub2c8\uub2e4.", "code": "error"}jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ curl "http://127.0.0.1/game/guess" -d "guess=50"
{"msg": "50, Answer : smaller", "code": "success"}jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ curl "http://127.0.0.1/game/guess" -d "guess=25"
{"msg": "25, Answer : smaller", "code": "success"}jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ curl "http://127.0.0.1/game/guess" -d "guess=10"
{"msg": "10, Answer : smaller", "code": "success"}jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ curl "http://127.0.0.1/game/guess" -d "guess=5"
{"msg": "5, Answer : greater", "code": "success"}jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ curl "http://127.0.0.1/game/guess" -d "guess=7"
{"msg": "7, Answer : greater", "code": "success"}jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$ curl "http://127.0.0.1/game/guess" -d "guess=9"
{"msg": "game clear in 8Trials", "code": "clear"}jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
jh_svt@jh-svt-ThinkPad-T470:/var/www/game$
```

게임시작성공 : {"code": "success"}

범위에 벗어난 값을 입력했을 경우 : {"msg": "You must enter a number over 0 and below **boundary**.", "code": "error"}

게임 시작후 입력 성공 :

{"msg": "**guess**, Answer : **smaller** or **greater**", "code": "success"}

# 블록코딩

initialize global URL to "http://172.20.10.3/game"

Url을 전역 변수로 설정 !

when Screen1 .Initialize  
do set btn\_go . Enabled to false

스크린 화면이 처음에 나왔을 때,  
Go버튼을 누르지 못하게 비활성화 시킴

when btn\_newgame .Click  
do set btn\_go . Enabled to false  
call no\_newgame .ShowTextDialog  
message "Enter number limit"  
title "Set Limit"  
cancelable true

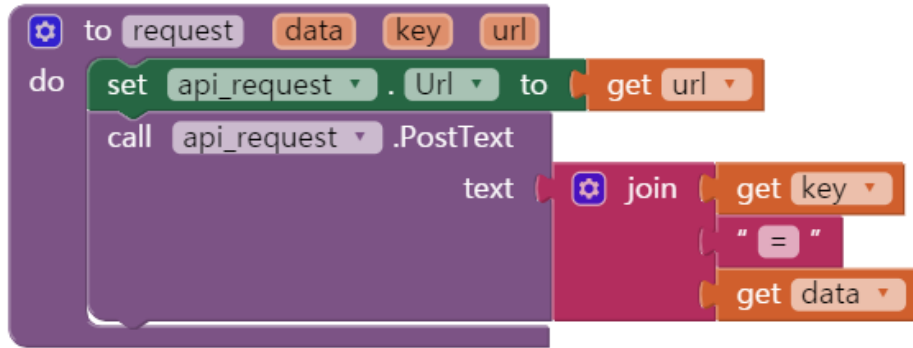
New Game 버튼이 클릭되면,  
Go 버튼은 비활성화 상태이고,  
Boundary 값을 입력할 창을 띄움 !

when no\_newgame .AfterTextInput  
response  
do call request  
data get response  
key "boundary"  
url join get global URL  
"/new"

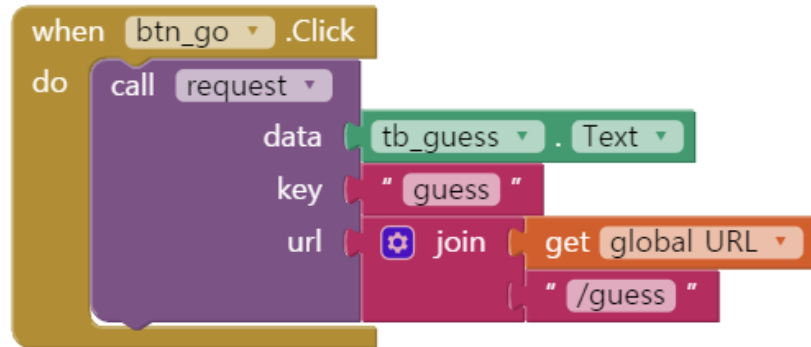
Boundary 값을 입력하고 확인 버튼을 누르면  
서버에 POST로 전역변수 url + /new 로  
Boundary 값을 보낸다



# 블록코딩



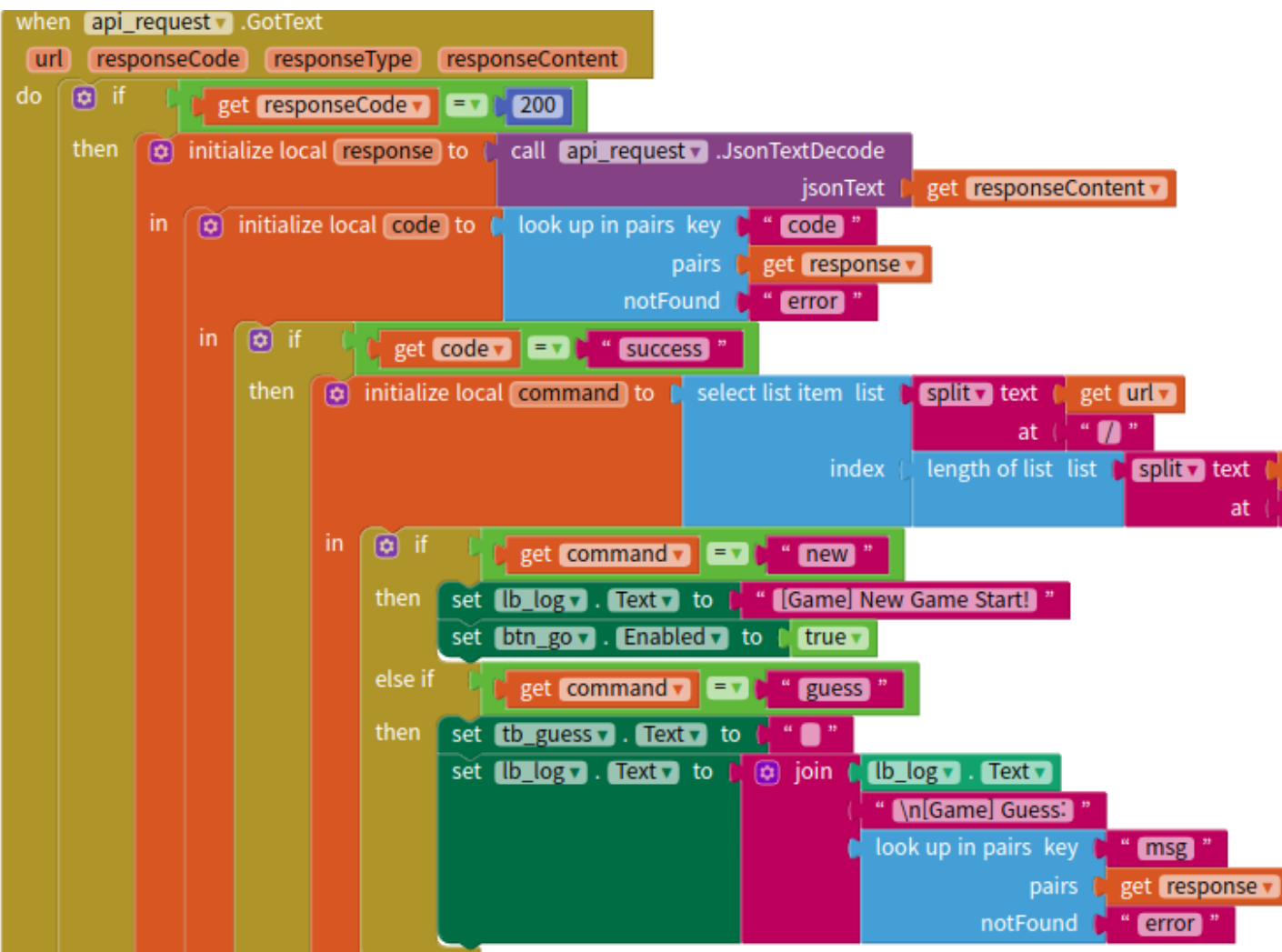
프로시저로 들어온 매개변수를  
웹 API 서버에 POST로 요청한다.



버튼 GO 가 클릭되면 텍스트박스에 입력한 수를  
전역 변수로 설정한 url + /guess 로 보낸다.

# 블록코딩

웹에서 responsecode = 200 이면  
Json으로 된 텍스트를 받아 decoding 해서,  
지역변수인 response 에 넣는다.



Response에 저장된 code에 해당하는 value 를  
지역변수 code에 저장한다.  
찾지 못하면 error 로 저장한다.

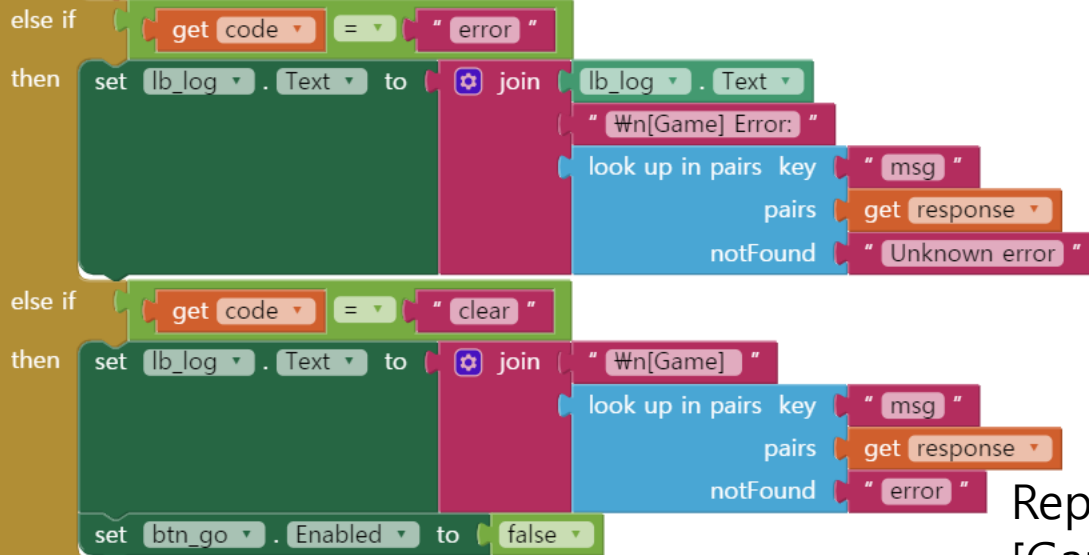
지역변수 code 에 저장된 값이 success일 때,  
요청 한 url이 어느 url인지 파악하기 위해  
Split을 이용해 나눠서 comman에 저장한다.

만약 command에 저장된 값이 new인 경우  
"[Game] New Game Start!"를 출력하고 Go버튼을  
활성화시킨다.

만약 command에 저장된 값이 guess 인 경우  
Textbox를 초기화 하고 guess의 결과를  
Lb\_log에 출력한다.

# 블록코딩

만약 code에 저장된 값이 error인 경우,  
[Game] Error와 respons의 msg 에 저장된 value를 띄운다.

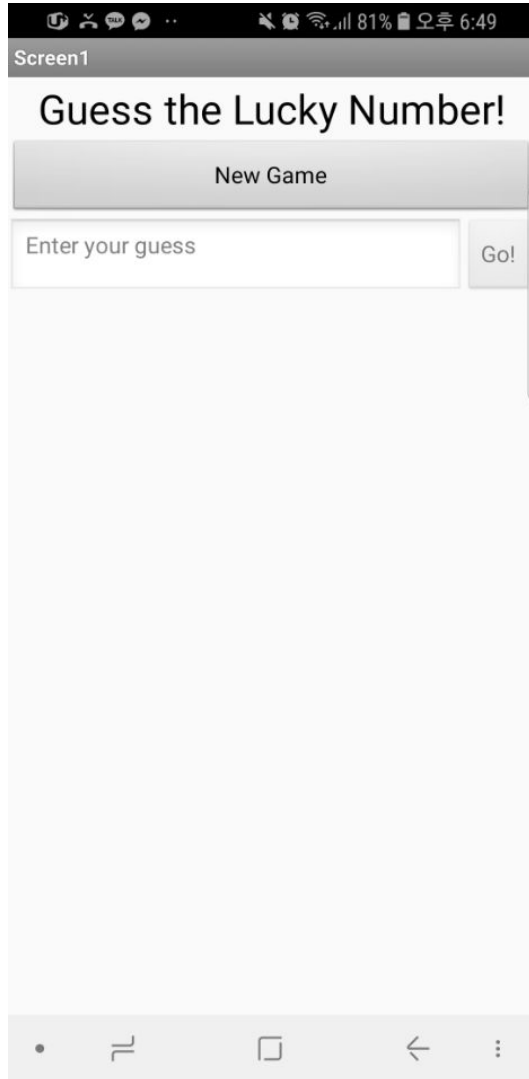


만약 code에 저장된 값이 clear인 경우,  
[Game]과 respons의 msg에 저장된 value를 띄우고,  
Go버튼을 비활성화 시킨다.

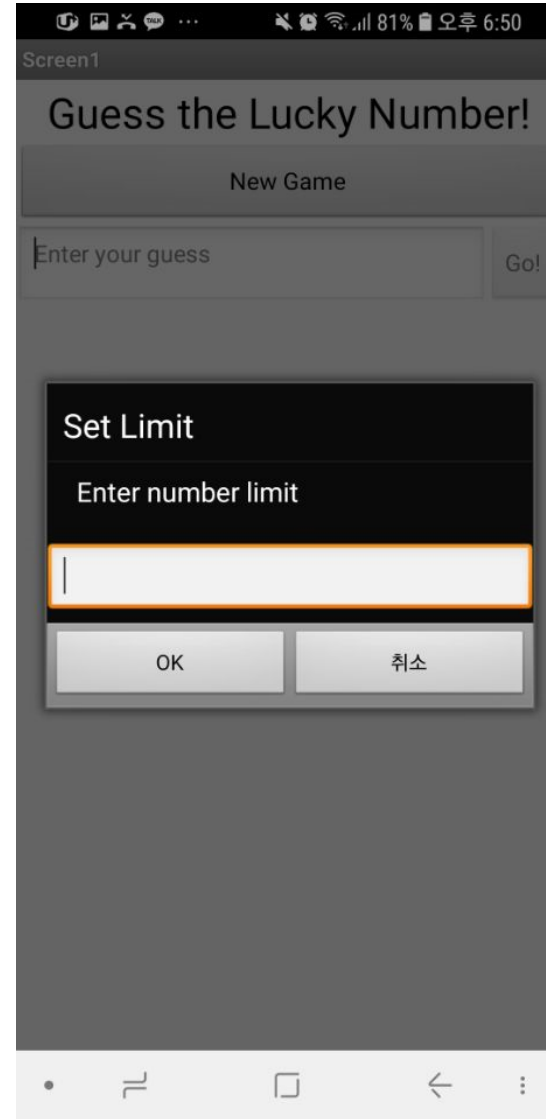
Reponsecode가 200이 아닐때 lb\_log에  
[Game] Error: 와 reponsecode에 저장된 값을 띄운다



# 클라이언트 UI 스크린 샷



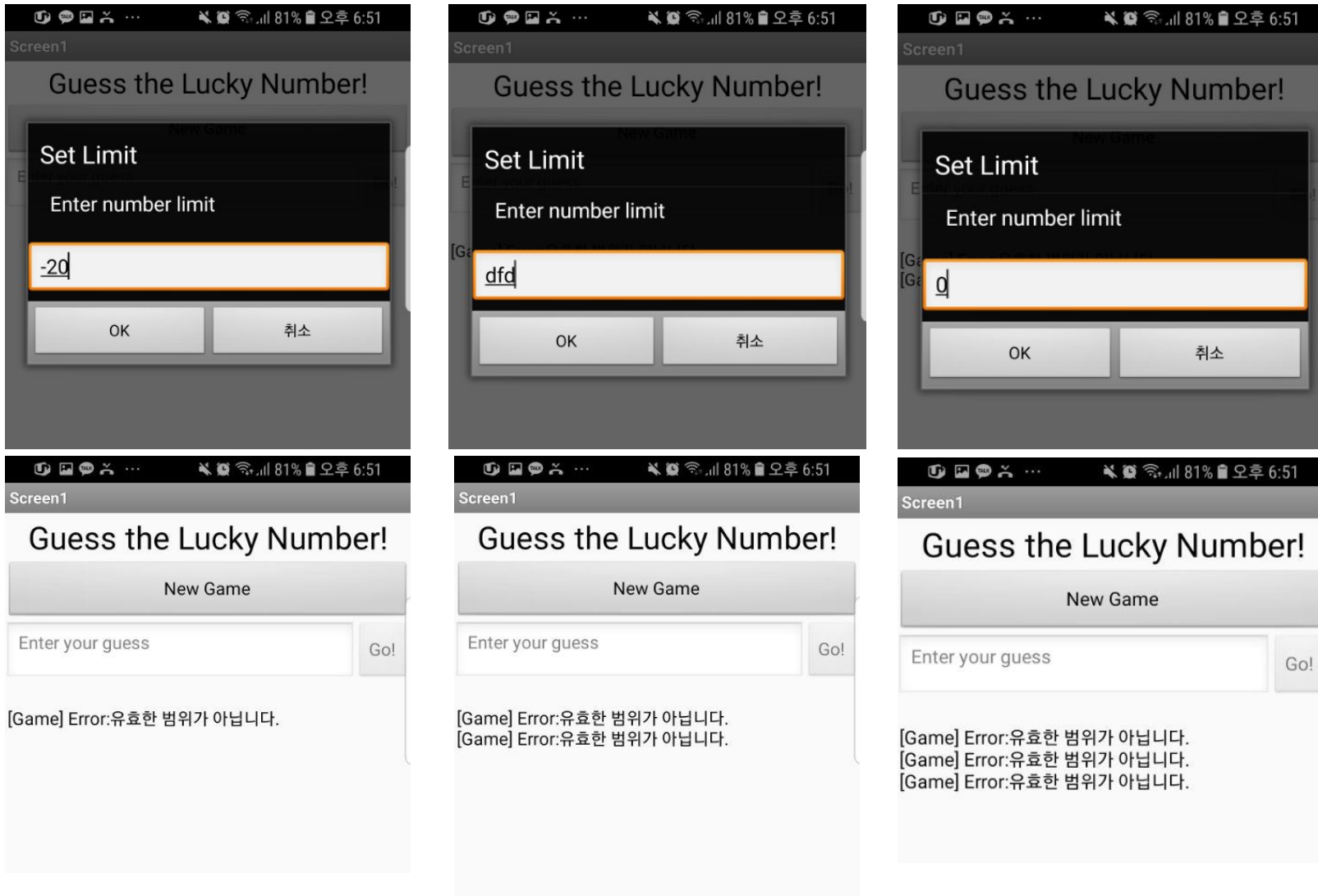
-> 앱을 켜면  
처음 나오는 화면  
GO 버튼은  
비활성화 되어있음.



-> New Game 버튼을  
누르면 나오는 화면.

boundary 값을  
입력할 수 있음.

# 클라이언트 UI 스크린 샷

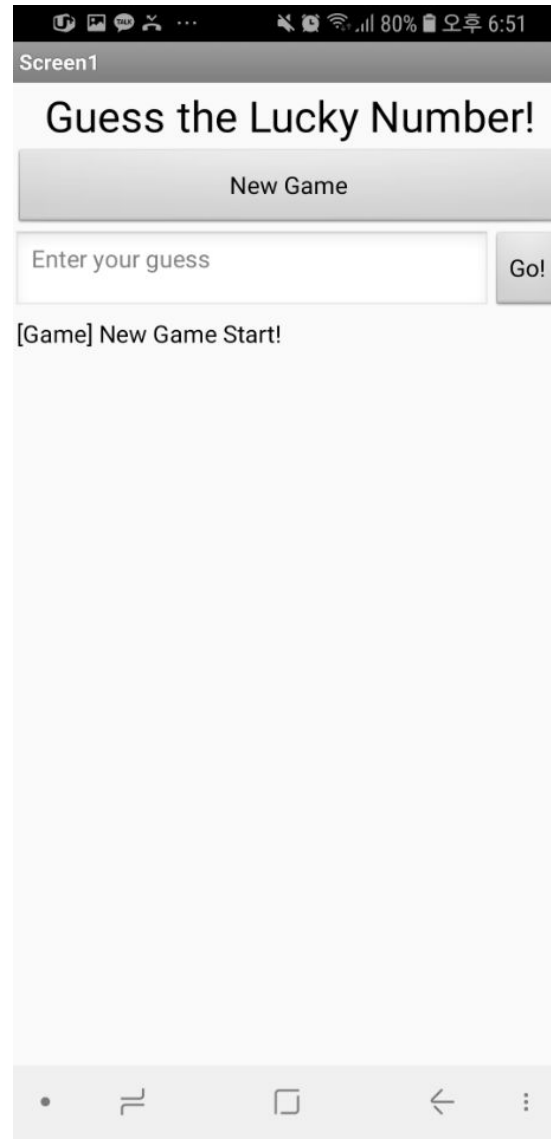
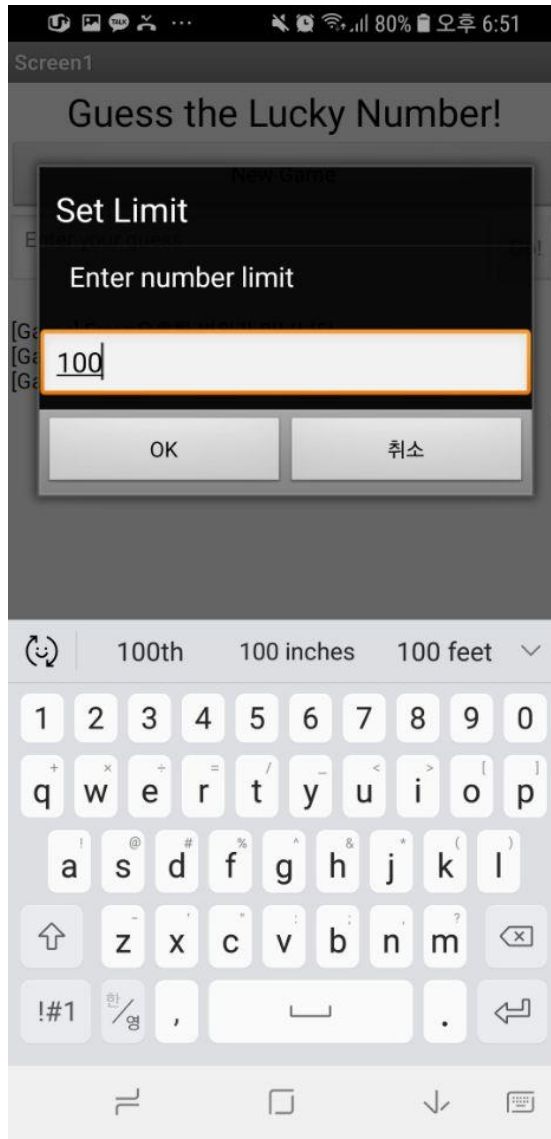


Boundary 값으로  
음수, 0, 문자열을 입력하면

"유효한 범위가 아닙니다."

라는 문구가 뜬.

# 클라이언트 UI 스크린 샷



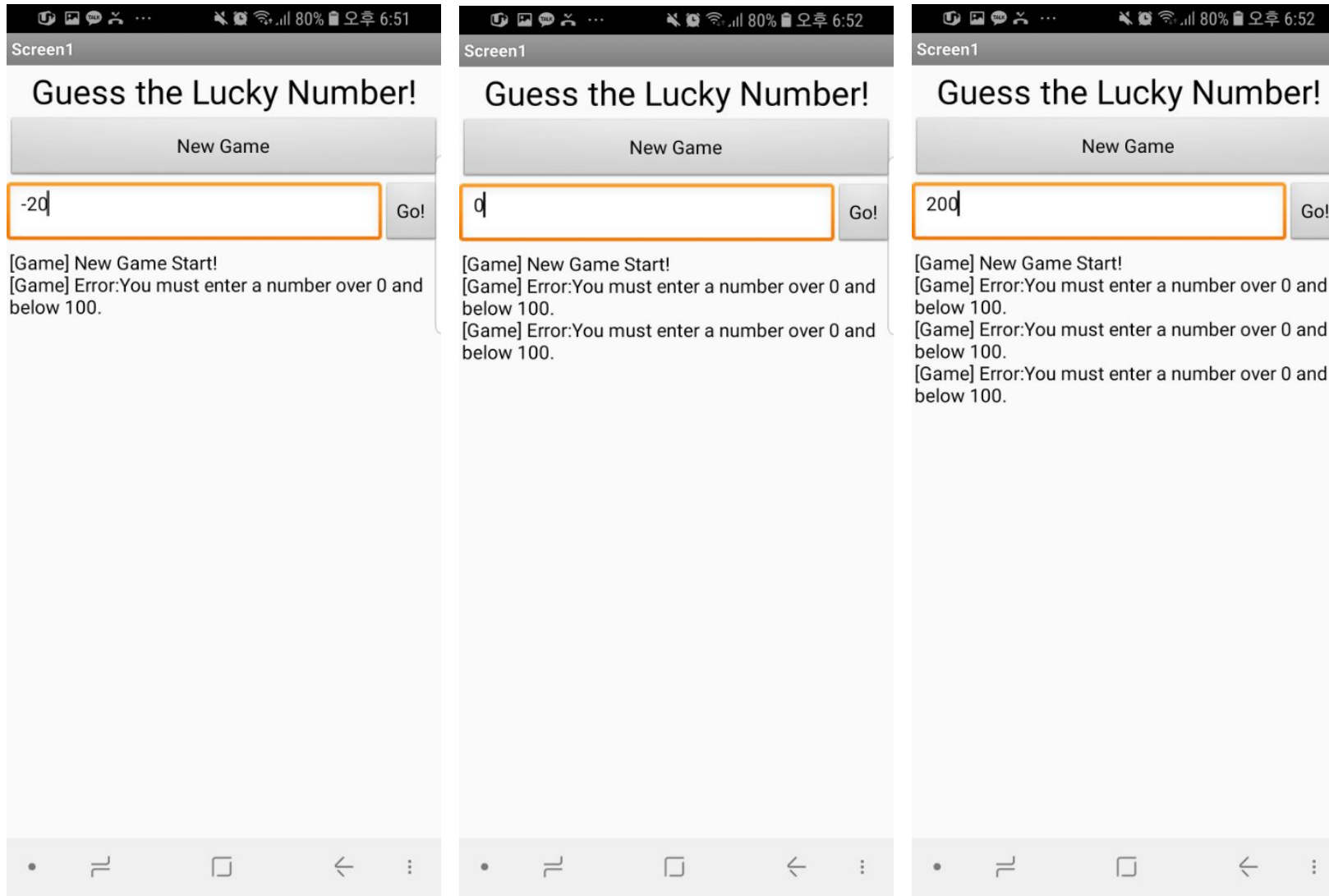
정상적인 값의  
Boundary를 입력하면

"New Game Start!"

라는 메시지와 함께  
게임이 시작되며,

Go 버튼이 활성화된다.

# 클라이언트 UI 스크린 샷



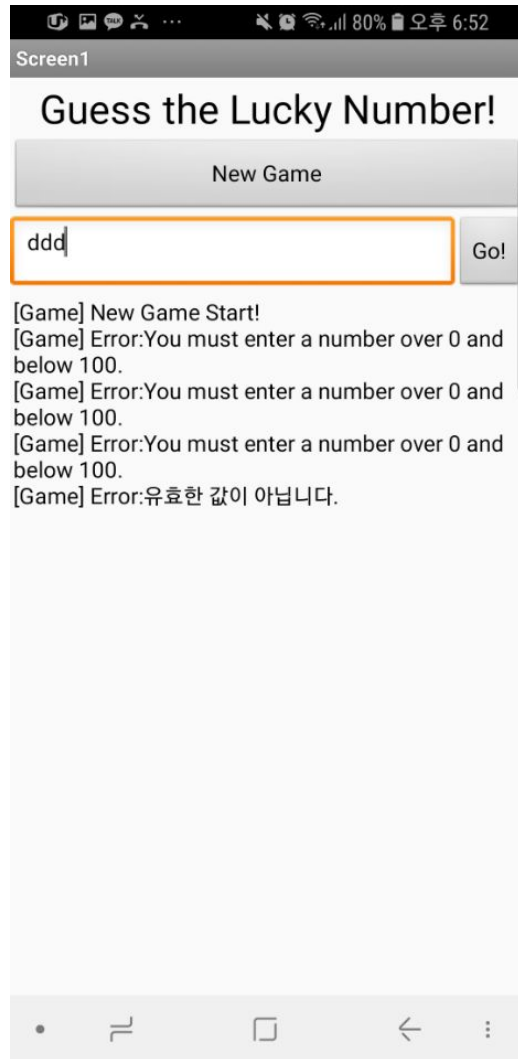
사용자가 guess값으로

1~ boundary에  
벗어나는 범위 값을  
입력하면

"You must enter a number  
Over 0 and below boundary"

라는 메시지가 출력됨.

# 클라이언트 UI 스크린 샷



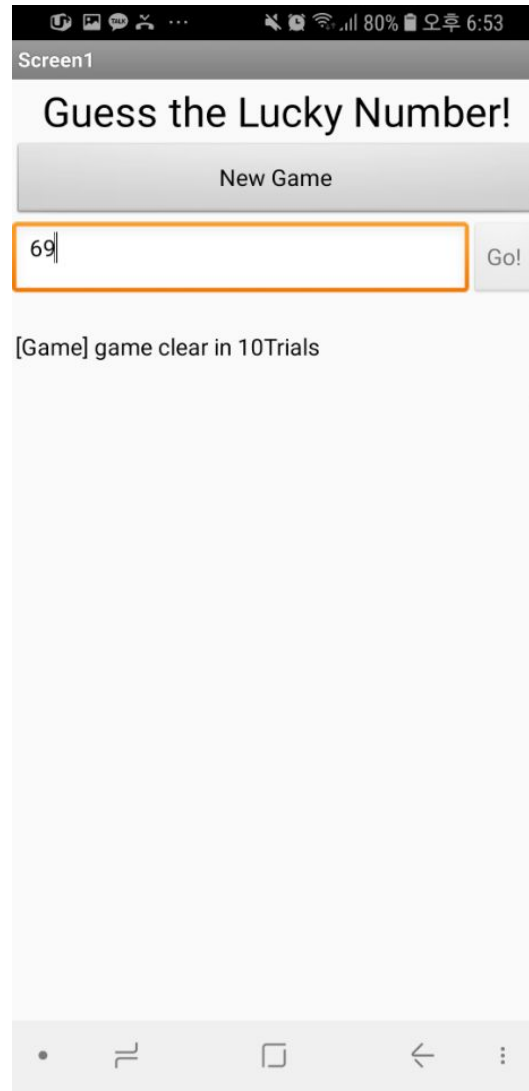
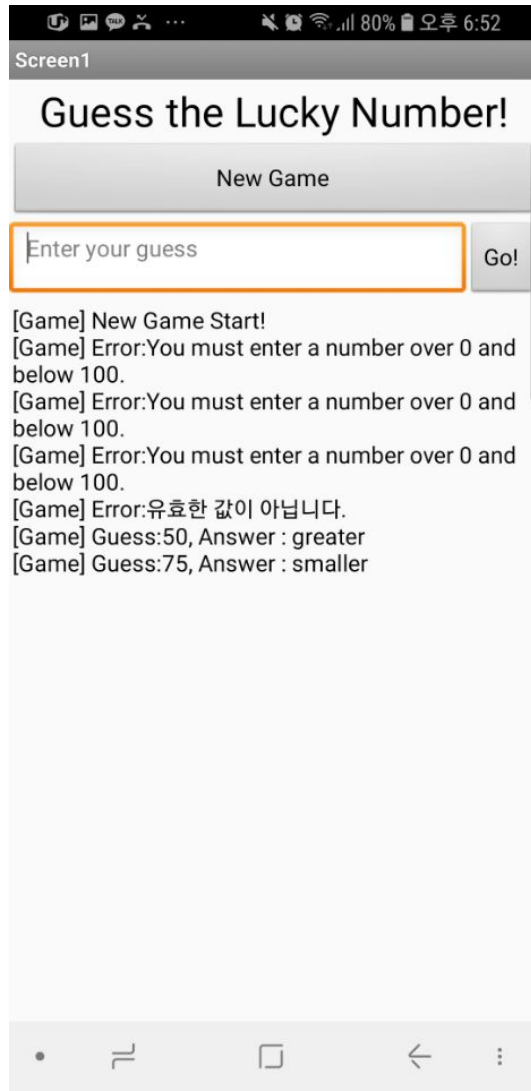
사용자가 guess에 문자열을 입력하면

“유효한 값이 아닙니다.”

라는 문구가 출력됨.



# 클라이언트 UI 스크린 샷



사용자의 입력값이 boundary보다 작으면

"greater" 이 출력되고

사용자의 입력값이 boundar보다 크면

"smaller" 이 출력되며

사용자가 답을 맞추면

"game clear in 'trials'Trials"  
가 출력되며 게임이 끝나고  
Go 버튼이 비활성화 된다.

감사합니다.