

소프트웨어 프로젝트 1

기말 프로젝트

해적 룰렛 게임 보고서

20181668

이예진

목차

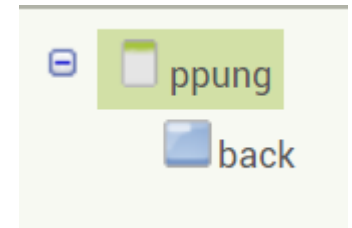
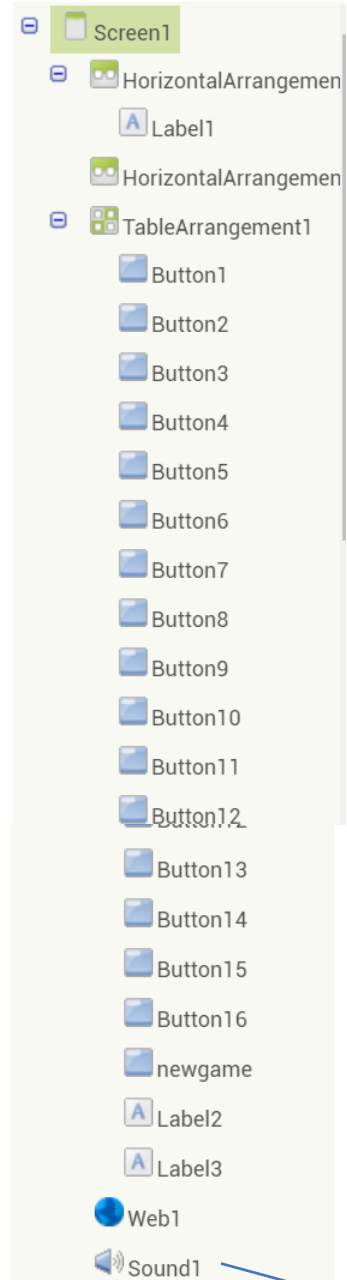
1. 앱의 목표
2. Client
3. API 설계 사항
4. Server
5. 테스트
6. 보완하고 싶은 부분

앱의 목표



1. 칼은 버튼으로 대체한다.(최대한 입력에 어려가 없도록)
2. 버튼은 한번 누르면 비활성화 시킨다.
3. 랜덤으로 당첨 번호를 지정한다.
4. 당첨 번호와 같은 버튼을 클릭하면 진동이 울리며 새로운 스크린을 띄어 걸렸음을 나타낸다.
5. 새로운 스크린에 있는 '돌아감' 버튼을 클릭하면 초기 화면으로 이동한다.
6. 라즈베리 파이를 활용하여 서버를 구현한다.

Client



진동 울리게 하기위해

initialize global url to "http://192.168.0.35:8888/"
 initialize global gameover to false

url과 gameover
 전역 변수생성

when Screen1.Initialize
 do
 set Button1.Enabled to false
 set Button2.Enabled to false
 set Button3.Enabled to false
 set Button4.Enabled to false
 set Button5.Enabled to false
 set Button6.Enabled to false
 set Button7.Enabled to false
 set Button8.Enabled to false
 set Button9.Enabled to false
 set Button10.Enabled to false
 set Button11.Enabled to false
 set Button12.Enabled to false
 set Button13.Enabled to false
 set Button14.Enabled to false
 set Button15.Enabled to false
 set Button16.Enabled to false

screen1

화면 처음 들어갔을 때
 숫자 버튼 비활성화

when Button1.Click
 do
 set Web1.Url to join get global url
 "guess/?guess=1"
 call Web1.Get
 if get global gameover = false
 then set Button1.Enabled to false

button1클릭하면 guess에 알맞게 url 설정
 gameover가 아니라면 button1 비활성화

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

web에서 얻어온 값과 입력
 한 key값과 동일한 게 있다
 면 encoded값을 디코딩하
 여 리턴하고 없다면 "Not
 found" 리턴

when newgame.Click
 do
 set Button1.Enabled to true
 set Button2.Enabled to true
 set Button3.Enabled to true
 set Button4.Enabled to true
 set Button5.Enabled to true
 set Button6.Enabled to true
 set Button7.Enabled to true
 set Button8.Enabled to true
 set Button9.Enabled to true
 set Button10.Enabled to true
 set Button11.Enabled to true
 set Button12.Enabled to true
 set Button13.Enabled to true
 set Button14.Enabled to true
 set Button15.Enabled to true
 set Button16.Enabled to true
 set Web1.Url to join get global url
 "new"
 call Web1.Get

newgame을 클
 릿하면 숫자버튼
 활성화
 url new에 알맞게
 설정

screen ppung

when back.Click
 do open another screen screenName "Screen1"

back버튼 누르면 screen1으로 다시 이동하게

when Web1.GotText
 url responseCode responseType responseContent
 do
 if get responseCode = 200
 then
 if call extract encoded get responseContent
 key "code" = "success"
 then
 if call extract encoded get responseContent
 key "gameover" = true
 then
 call Sound1.Vibrate
 millisecs 1000
 set global gameover to true
 open another screen screenName "ppung"
 else
 set Label2.Text to "Network Error"
 set Label3.Text to get url

200이고 code가 success이고 gameover가 true이면
 당첨번호가 걸린 것이므로 진동이 울리고 gameover
 변수가 true로 바뀌며 ppung스크린으로 이동한다.
 만약 200이 아니라면 network error이므로 url를 출
 령하도록 한다.

API 설계 사항

- 새로운 게임의 시작
 - URL - `http://192.168.0.35/new`
 - 출력 형태(json)
`{'code': 'success'}`
- 숫자 추측
 - URL - `http://192.168.0.35/guess`
 - 입력 인자
 - guess : 사용자가 맞추는 것을 시도
ex) 버튼 1을 눌렀을 때-> `http://192.168.0.35/guess?guess=1`
 - 출력 형태(json)
`{'a': guess, 'code': 'success', 'num': a.gameover, 'trials': trials, 'secre': c}`
- 둘 다 아니라면?
 - 출력 형태(json)
`{'code': 'error', 'msg': 'non-existent API method'}`

server

WSGI Application wsgi.py

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

def application(environ, start_response):
    response = {'code': 'error', 'msg': 'wrong HTTP method'}
    path = environ['PATH_INFO'].split('/')
    method = path[1]
    if method == 'new':
        response = gamedriver.new_game()
    elif method == 'guess':
        d = parse_qs(environ["QUERY_STRING"])
        guess = escape(d.get('guess', [''])[0])
        guess = int(guess)
        response = gamedriver.guess(d)
    else:
        response = {'code': 'error', 'msg': 'non-existent API method'}

    status = '200 OK'
    response_body = json.dumps(response)
    response_headers = [
        ('Content-Type', 'application/json'),
        ('Content-Length', str(len(response_body)))
    ]
    start_response(status, response_headers)
    return [response_body]

httpd = make_server('localhost', 'Port', application)
httpd.serve_forever()
```

game Driver gamedriver.py

```
from tongman import Tongman
a = Tongman()

def new_game():
    a.newGame()
    return {'code' : 'success'}

def guess(d):
    guess = d.get('guess', [''])[0]
    b = a.guess(guess)
    c = a.secretnum
    trials = a.tried()
    return {'a':guess, 'code': 'success', 'num': a.gameover, 'trials': trials, 'secre':c}
```

game Logic tongman.py

```
import random

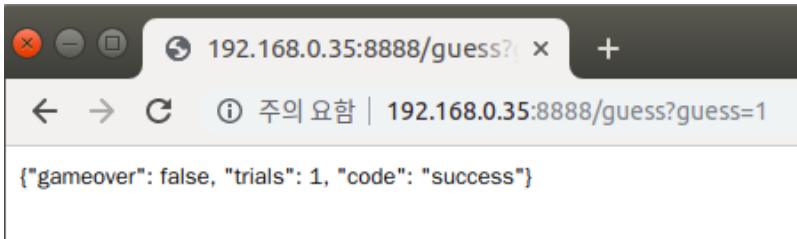
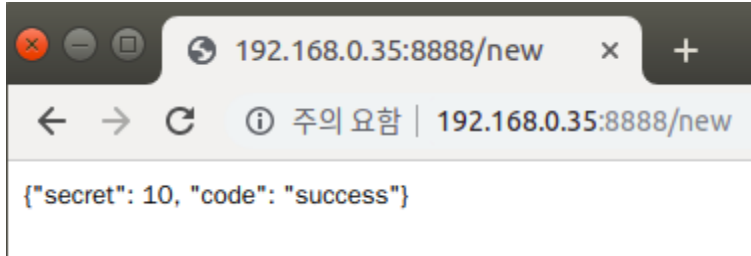
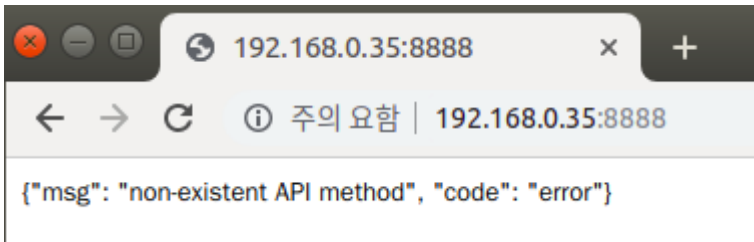
class Tongman:
    def __init__(self):
        self.secretnum = 0
        self.trials = 0
        self.gameover = False

    def newGame(self):
        self.secretnum = random.randint(1,16)
        self.trials = 0
        self.gameover = False

    def guess(self, userGuess):
        if self.secretnum == int(userGuess):
            self.trials += 1
            self.gameover = True
        else:
            self.trials += 1

    def tried(self):
        return self.trials

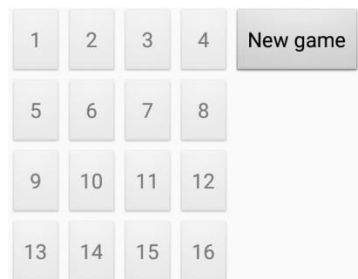
if __name__ == "__main__":
    a = Tongman()
    a.newGame()
    while a.gameover != True:
        uinput = int(input())
        a.guess(uinput)
    print(a.tried())
```



추측 시도
추측 성공
시도 횟수

```
pi@raspberrypi:~ $ cd swp1/  
pi@raspberrypi:~/swp1 $ python tongman.py  
3  
8  
2  
3
```

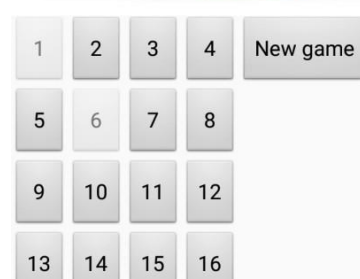

테스트



초기 상태



new game을 눌렀을 때



번호 눌렀을 때



걸렸다!



돌아가기 버튼 클릭

보완하고 싶은 부분

버튼이 16개나 되니 블록 코드가 너무 지저분하다. 이 부분은 고치고 싶다. 그리고 최대한 예외가 나오지 않도록 고려하였지만 그래도 예외는 있을 텐데 이 부분에 대해 더욱 보완하고 싶다.

server코드에 대해 직접 보고 싶다면 <https://github.com/yeahjin/SWP1AD>