

Lab #9- Baseball Game (baseball_game)

Copyright 2015 © document created by TeamLab.Gachon@gmail.com
(<mailto:TeamLab.Gachon@gmail.com>)

Introduction

수많은 난간. 자그마치 8개의 Lab을 끝내고 여기까지 온 여러분들을 환영한다. 이 Lab은 사상 최초로 어렵다. 이때까지 어려운 Lab은 없었다. 단지 조금 시간이 오래 걸리거나 약간 복잡하게 느껴졌을 뿐이다. 근데 이번엔 어렵다. 정말이다. 하지만 어려운 만큼 가장 여러분에게 도움이 많이 되는 숙제라고 믿는다. 참고로 2014년 수강생 설문조사에서도 가장 어렵지만 가장 도움이 많이된 숙제로 뽑혔다.

본 Baseball Game Lab은 간단한 숫자 맞추기 게임이다. 컴퓨터는 3자리의 Random Number를 생성하고 사용자는 숫자를 입력하면, 그 세자리 숫자가 Random Number와 얼마나 비슷한지 Strike과 Ball로 알려주는 게임이다. 간단한 규칙은 다음과 같다.

- 사용자가 입력한 숫자와 컴퓨터가 생성한 숫자의 한 숫자와 자릿수가 모두 일치하면 1 Strike
- 자릿수는 다르나 입력한 한 숫자가 존재하면 1 Ball
- 세자리 숫자를 정확히 입력하면 3 Strike로 게임 종료

이 때 컴퓨터가 생성하는 세 자리 숫자는 반드시 같은 숫자를 한 개만 가지고 있어야 한다. 예를 들면 332, 474, 555 처럼 같은 숫자가 2개 이상있는 숫자는 게임에서 제외되어야 한다. 실제 볼 스트라이크 판정 예제는 다음과 같다.

컴퓨터 생성 숫자	사용자 입력	판정 결과
472	123	0 Strike, 1 Ball
	547	0 Strike, 2 Ball
	247	0 Strike, 3 Ball
	742	1 Strike, 2 Ball
	472	3 Strike

어떻게 할지 걱정될 수도 있지만, 꽤 많은 힌트와 함께 여러분들을 바른 길로 인도해주는 함수를 제시할 것이다. 너무 걱정말고 시작해보자

Baseball Game Overview

단순히 Strike와 볼판정만 해주면 좋겠지만 본 Lab에서는 처음으로 사용자의 입력에 대한 오류 처리를 실시한다. 여러분들도 가끔 웹 사이트에 접속할 때 잘못된 입력을 바로 잡아주는 프로그램을 본적이 있을 것이다. 예를 들면, 이메일 주소를 입력할 때 “@”를 쓰지 않았다던가, 비밀번호 입력시 숫자를 같이 쓰지 않았다던가 할 경우, 자동으로 프로그램이 오류 입력임을 지정하고 재입력을 하게해 줄 것이다. 본 프로그램도 마찬가지로 사용자가 아래와 같이 잘못 입력했을 경우, “Wrong Input” 이라는 메시지를 남기며 오류 처리를 해준다.

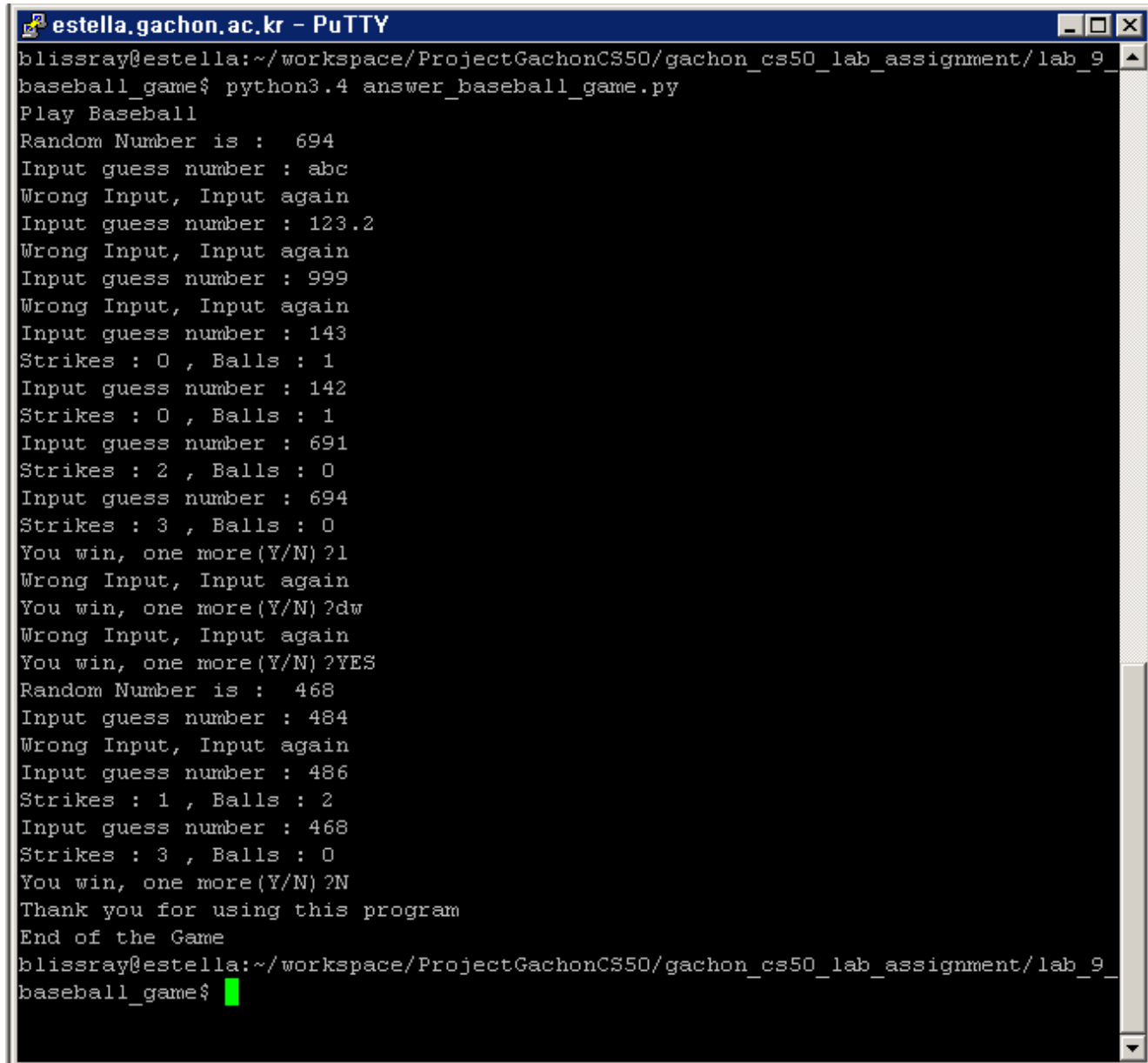
- 사용자가 숫자 대신 문자를 입력했을 경우
- 사용자가 실수를 입력했을 경우
- 사용자가 입력한 세 자리 정수에 중복되는 숫자가 있을 경우

또한 사용자가 숫자를 정확히 입력하여 “3 Strikes” 나왔을 경우, 사용자가에게 한번 더 게임을 할 의향이 있는지 물어보고 의향이 있을 경우 다시 게임을 실행 시킨다. 이 때 아래와 같은 조건으로 게임을 재시작을 결정한다.

- 사용자가 “Y” 또는 “Yes”라고 입력했을 경우 대소문자에 상관없이 게임을 재시작한다. 즉 “y”, “yEs”, “yes” 등의 경우에도 게임은 다시 시작된다.
- 사용자가 “N” 또는 “No”라고 입력했을 경우 대소문자에 상관없이 게임을 종료한다. 즉 “n”, “nO”, “NO” 등의 경우에도 게임은 종료된다.
- 사용자가 이외에 입력을 하였을 경우 “Wrong Input” 이라고 출력되면서 종료된다.

또 다른 종료 조건으로 사용자가 게임 중 0 를 입력하게 되면 종료해야 한다. 이는 Lab 8의 Factorial Calcualtor의 종료 조건과 같다.

또한 새로운 게임이 시작할 때는 반드시 Random Number is : xxx 라는 문자가 출력되면서 xxx 부분에는 컴퓨터가 random하게 생성한 것을 표시해준다. 이는 자동 채점 프로그램을 동작하게 하기 위한 것으로 표시 되지 않을 경우 컴퓨터가 채점을 하지 못한다. 실제 돌아가는 프로그램은 아래와 같다.



```
estella.gachon.ac.kr - PuTTY
blissray@estella:~/workspace/ProjectGachonCS50/gachon_cs50_lab_assignment/lab_9_
baseball_game$ python3.4 answer_baseball_game.py
Play Baseball
Random Number is : 694
Input guess number : abc
Wrong Input, Input again
Input guess number : 123.2
Wrong Input, Input again
Input guess number : 999
Wrong Input, Input again
Input guess number : 143
Strikes : 0 , Balls : 1
Input guess number : 142
Strikes : 0 , Balls : 1
Input guess number : 691
Strikes : 2 , Balls : 0
Input guess number : 694
Strikes : 3 , Balls : 0
You win, one more(Y/N)?l
Wrong Input, Input again
You win, one more(Y/N)?dw
Wrong Input, Input again
You win, one more(Y/N)?YES
Random Number is : 468
Input guess number : 484
Wrong Input, Input again
Input guess number : 486
Strikes : 1 , Balls : 2
Input guess number : 468
Strikes : 3 , Balls : 0
You win, one more(Y/N)?N
Thank you for using this program
End of the Game
blissray@estella:~/workspace/ProjectGachonCS50/gachon_cs50_lab_assignment/lab_9_
baseball_game$
```

```

estella.gachon.ac.kr - PuTTY
blissray@estella:~/workspace/ProjectGachonCS50/gachon_cs50_lab_assignment/lab_9_
baseball_game$ python3.4 answer_baseball_game.py
Play Baseball
Random Number is : 572
Input guess number : 58
Wrong Input, Input again
Input guess number : 582
Strikes : 2 , Balls : 0
Input guess number : 571
Strikes : 2 , Balls : 0
Input guess number : 573
Strikes : 2 , Balls : 0
Input guess number : 0
Thank you for using this program
End of the Game
blissray@estella:~/workspace/ProjectGachonCS50/gachon_cs50_lab_assignment/lab_9_
baseball_game$

```

숙제 template 파일 다운로드

먼저 숙제 template 파일을 cs50 서버로 부터 다운로드 받는다. 로그인 후 나타나는 bash shell 에서 다음과 같은 명령을 입력하자.

```
python3.4 submit_assignment.py -get baseball
```

입력되면 다운로드 안내 메시지와 함께 baseball_game.py 파일이 다운로드 된다.

baseball_game.py 파일 Overview

vim editor 로 baseball_game.py 을 열어 전체적인 개요를 보자. vi baseball_game.py 명령으로 파일을 열어보면 main 함수와 여러개의 함수들이 존재할 것이다. 각 함수들은 여러분께서 직접 작성해서 제출해야 하는 함수이고, main 함수는 실제 baseball game을 실행하는 함수이다. 각 함수의 구현 내용은 아래와 같다.

함수	설명
is_digit	문자열 값을 입력받아, 정수로 변환 가능할 경우는 True, 그렇지 않을 경우는 False로 변환해주는 함수
	문자열 값을 입력받아, 정수로 변환하여 100이상 1000미만일 경우 True, 그렇지 않을

is_between_100_and_999	경우는 False를 반환해주는 함수. 이 때 입력되는 문자열 값은 정수형태의 문자열 값임이 보장됨 ex) 324, 1103
is_duplicated_number	문자열로 된 세자리 양의 정수 값을 입력 받아 정수로 변환하였을 경우 중복되는 수가 있으면 True, 그렇지 않을 경우는 False를 반환해주는 함수. 이 때 문자열로 된 세자리 양의 정수값의 입력이 보장됨 ex) 117 - True, 123 - False, 103 - False, 113 - True
is_validated_number	문자열 값을 입력받아 아래 세가지 조건에 맞으면 True, 그렇지 않으면 False를 반환함 1) 숫자형 문자열이며, 2) 100이상 1000미만이며, 3) 중복되는 숫자가 없을 경우. 위의 세 가지 함수를 사용하여 구현이 가능함
get_not_duplicated_three_digit_number	입력 값은 없으며, 중복되는 숫자가 없는 세자리 정수값이 반환됨, 문자열이 아닌 정수값으로 반환됨. 이때 get_random_number() 함수를 사용하여 random number를 생성하고 생성된 random number에 중복되는 숫자가 없을 경우에만 해당 숫자를 반환해줌
get_strikes_or_ball	사용자가 입력한 세 자리 정수 문자열과, 컴퓨터가 자동으로 생성한 세 자리 정수 문자열을 입력받아, baseball game 규칙에 맞춰 [strikes, balls]을 반환해줌. 리스트 형태로 반환하여 strikes와 balls는 Integer Type 값
is_yes	문자열값을 입력받아 해당 입력값이 대소문자에 관계없이 "Y" 또는 "YES"일 경우 True, 그렇지 않을 경우 False를 반환함
is_no	문자열값을 입력받아 해당 입력값이 대소문자에 관계없이 "N" 또는 "NO"일 경우 True, 그렇지 않을 경우 False를 반환함

상당히 많고 복잡하고, 게다가 어렵기까지 하다. 일단 하나하나 해결해보길 바란다.

참고로 get_random_number() 함수는 임의의 3자리 자연수를 반환하는 함수로 반드시 써야 되는 함수니 get_not_duplicated_three_digit_number 함수에서 사용하기 바란다.

main 함수 수정하기

위의 함수도 상당히 어렵지만 본 Lab의 가장 어려운 숙제는 바로 main 함수를 수정하는 일이다. main 함수의 기본 template은 아래와 같다.

```
def main():
    print("Play Baseball")
    user_input = 999
    random_number = str(get_not_duplicated_three_digit_number())
    print("Random Number is : ", random_number)
    # ===Modify codes below=====
    # 위의 코드를 포함하여 자유로운 수정이 가능함

    # =====
    print("Thank you for using this program")
    print("End of the Game")
```

template은 lab 8 factorial calculator와 흡사하면서도 조금 다르다. 일단 프로그램이 시작되면 “Play Baseball” 한줄과 “Random Number is :” 라는 한 줄이 화면에 찍히고 시작한다. 앞에서 설명했듯이 반드시 새로운 baseball game이 시작되면 `print("Random Number is : ", random_number)` 가 실행되어야 한다. `random_number` 변수는 수강생이 임의로 수정가능하겠지만 `get_not_duplicated_three_digit_number` 함수를 잘 작성하여 `random_number` 에 할당해 주는게 이 숙제를 해결하는데 있어 가장 쉬울 것이다.

이 숙제가 까다로운 것은 숙제 자체도 그렇지만, 숙제를 검사할 때 검사 프로그램의 한계로 인해 위쪽 screen shot 처럼 작동되지 않으면 올바르게 코딩했음에도 불구하고 pass에 실패할 수 있다.

한 가지 유의할 점은 사용자의 입력을 받을 때 `print` 문을 쓴 다음 `input` 문을 쓰지 말고 `input` 문만 사용하여 입력을 받아야 한다. 예를 들면 `user_input = input('Input guess number : ')` 형태로 작성되어야 한다. 이 또한 강의자의 개발 능력 한계니 이해해주면 좋겠다. 먼저 각 개별 함수를 목적에 맞게 잘 수정한 후, 개별 함수들을 잘 활용하여 `main` 함수를 작성하기 바란다. 실패할 가능성이 높은 함수지만, 굉장히 즐거운 작업이 될 것이다.

숙제 제출하기

모든 함수를 다 수정했다면, 아래와 같이 제출하자

```
python3.4 submit_assignment.py -submit baseball_game.py
```

제대로 작성했다면 아래와 같은 메시지가 뜰 것이다.

-----	-----	-----
Function Name	Passed?	Feedback
-----	-----	-----
is_digit	PASS	Good Job
is_yes	PASS	Good Job
is_duplicated_number	PASS	Good Job
is_no	PASS	Good Job
get_strikes_or_ball	PASS	Good Job
is_validated_number	PASS	Good Job
is_between_100_and_999	PASS	Good Job
main	PASS	Good Job
get_not_duplicated_three_digit_number	PASS	Good Job
-----	-----	-----

아마도 정말로 감격 스러운 순간일 것이다. 고생했다.

Next Work

이제 여러분은 프로그래밍을 위한 거대한 한발을 내딛었다고 생각한다. 이때까지는 A,B, C를 부르거나 기껏해야 “How are you? Fine thank you, and you” 정도를 읊었다면 이제 프로그래밍으로 쉬운 어린이 “팝송” 한국 부른 정도라고나 할까? 여러분들은 충분히 잘했다. 오늘 자기전에 맥주한잔을 마시면서 자신이 무한한 가능성이 있는 존재임을 깨닫기 바란다. 여기까지 와줘서 고맙다.

Human knowledge belongs to the world - from movie ‘Password’ -

Footnotes