

2023년 2학기

프로그래밍과 문제해결

Assignment #1

담당교수: 윤은영

학번: 20230024

학과: 무은재학부

이름: 문요준

POVIS ID: yojun313

개발 환경: Mac OS VS CODE

명예서약(Honor code)

“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

Problem: 목찌빠 계단오르기

코드를 Mac OS VS CODE로 작성하였습니다. Windows 환경과 차이가 존재합니다.

- 화면지우기 시 Windows에서는 `os.system('cls')`, Mac에서는 `os.sytstem('clear')`
- 특수기호가 차지하는 영역 차이

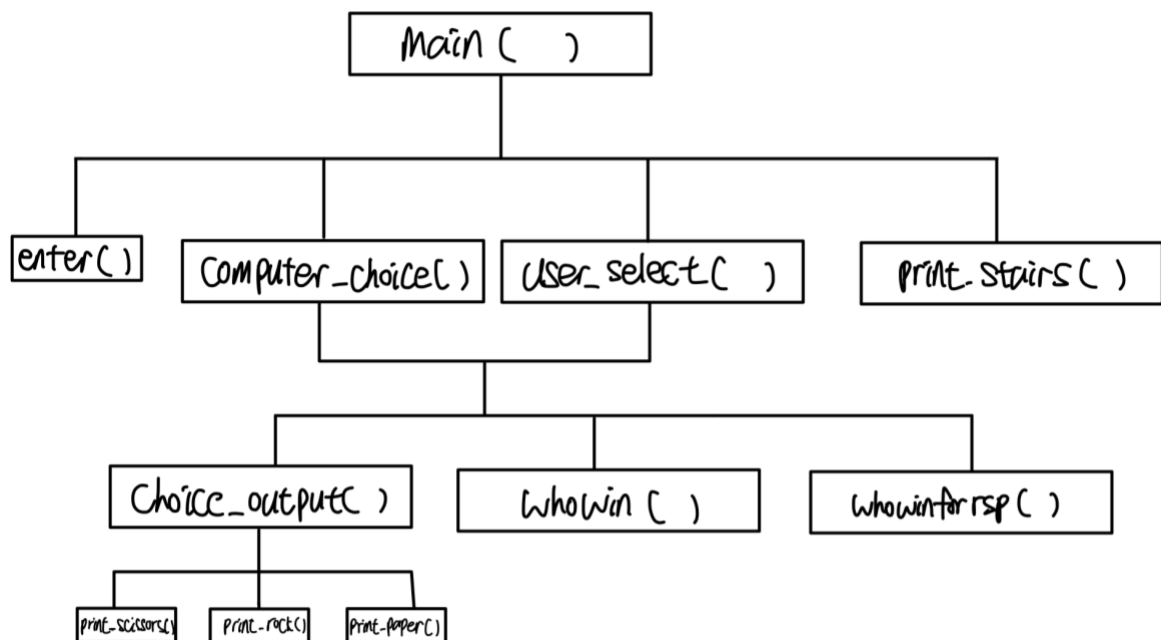
Mac OS에서 제작한 코드를 제출하겠습니다.

1. 문제의 개요

본 프로그램을 간략시 설명하면 다음과 같다.

- USER와 COMPUTER가 공격권 결정 가위바위보를 해 목찌빠 공격권을 정하고 목찌빠를 한다.
- USER는 가위, 바위, 보 중에 하나를 입력하고 COMPUTER는 랜덤으로 세 개의 옵션 중 하나를 고르고 출력함으로써 게임을 진행한다.
- 목찌빠에서 승패 판정이 나지 않으면 이동하는 계단의 수를 1칸씩 증가시킨다.
- COMPUTER 또는 USER가 먼저 반대편 계단의 끝으로 이동하면 승리한다.

[Structure Chart]



함수명	기능
-----	----

enter()	터미널에 출력된 텍스트를 os.system('clear') (맥 OS 기준)으로 삭제한다.
choice_output()	COMPUTER와 USER의 결정을 parameter로 받고 밑 세 개 함수를 실행한다
print_scissors()	가위 모양을 텍스트로 출력한다.
print_rock()	바위 모양을 텍스트로 출력한다.
print_paper()	보 모양을 텍스트로 출력한다.
computer_choice()	COMPUTER의 선택을 랜덤으로 결정하고('가위', '바위', '보' 중) 이를 반환한다.
user_select()	USER의 선택을 입력받고 이를 반환한다.
who_win()	COMPUTER와 USER의 선택을 parameter로 받고 묵찌빠의 승패를 판정하는 함수이다. 전역변수 win_num가 1일 때는 USER가 묵찌빠의 공격권을, win_num가 2일 때는 COMPUTER가 묵찌빠의 공격권을 갖고있는 상황이다. win_num가 1또는 2일 때 각각의 상황에 대해서 USER와 COMPUTER의 선택이 같으면 전역변수 win_num을 4(USER가 묵찌빠 승), 5(COMPUTER가 묵찌빠 승)로 변경한다. 선택이 다르면 win_num을 변경하지 않고 whowinforrsp 함수를 호출하고 이동 칸수를 1 증가시킨다.
whowinforrsp()	공격권 결정 가위바위보, 묵찌빠에서 승패판정이 나지 않았을 때(두 Player의 선택이 다를 때) 사용되는 함수이다. USER와 COMPUTER의 선택을 parameter로 받고 각각의 상황에 대해서 누가 공격, 수비인지를 출력하고 USER가 이겼을 때 전역변수 win_num을 1로, COMPUTER가 이겼을 때 2로 변경한다.
print_stairs()	<p>총 계단 개수, USER가 이동한 계단 칸 수, COMPUTER가 이동한 계단 칸 수를 parameter로 받고 현재 상황을 출력하는 함수이다. 계단을 그리는 원리는 다음과 같다. (stairs_num = 총 계단 개수)</p> <ol style="list-style-type: none"> 1. 가로줄의 행이 증가할수록(0,1,2,...,i) 왼쪽 계단은 0,1,2,...,i으로 증가하는 경향을 보이므로 우선 가로줄 리스트에 왼쪽 계단 "▮"을 i개 만큼 추가한다. 2. 가로줄의 왼쪽 계단과 오른쪽 사이의 공백에 (stairs_num+1)-2*i개 만큼, " "을 추가한다. (스페이스바 한 칸) (맥 OS 기준 "▮"와 " "가 차지하는 영역은 동일) 3. 가로줄 리스트에 오른쪽 계단 "▮"을 i개 만큼 추가한다. 4. 세로줄 리스트에 방금 생성한 가로줄 리스트 1개를 추가하고 가로줄 리스트를 []로 초기화한다. 5. 매개변수로 받은 USER와 COMPUTER가 이동한 계단 칸수에 기반하여 좌표를 계산하고 2차원 리스트[좌표][좌표]에 입력하여 그 위치를 "○", "●"로 바꿔준다. 6. COMPUTER와 USER의 계단 위치가 겹치게 되면 그 위치는 "●"로 변경한다.

2. 알고리즘

본 프로그램 작성을 위한 알고리즘을 Pseudo 코드 형태로 나타내면 다음과 같다.

[Pseudo Code]

os, random 모듈 불러옴

print(처음 게임 시작 화면)

stairs_num 변수에 총 계단 개수 입력받음

os.system('clear')로 터미널 텍스트 삭제

print_stairs()로 현재 상태 출력

while(전체 게임이 끝날 때까지):

 while(공격권 결정 가위바위보가 끝날 때까지):

 com_choice 변수에 computer_choice 함수로 COMPUTER 결정 대입

 user_choice 변수에 user_select 함수로 USER 결정 대입

 choice_output()을 호출해 현재 상황 그림 출력

 print(whowinforrsp로 USER, COMPUTER 승패) 출력

 while(묵찌빠가 끝날 때까지):

 print(승리 시 이동 칸 수, move_num)

 whowinforrsp를 호출, USER와 COMPUTER의 승패를 나타내는 전역변수 win_num 변경

 com_choice 변수에 computer_choice 함수로 COMPUTER 결정 대입

 user_choice 변수에 user_select 함수로 USER 결정 대입

 choice_output()을 호출하여 현재 상황 그림 출력

 whowin()을 호출하여 win_num을 4또는 5변경 / 유지 후 이동 칸 수 증가(승패 X)

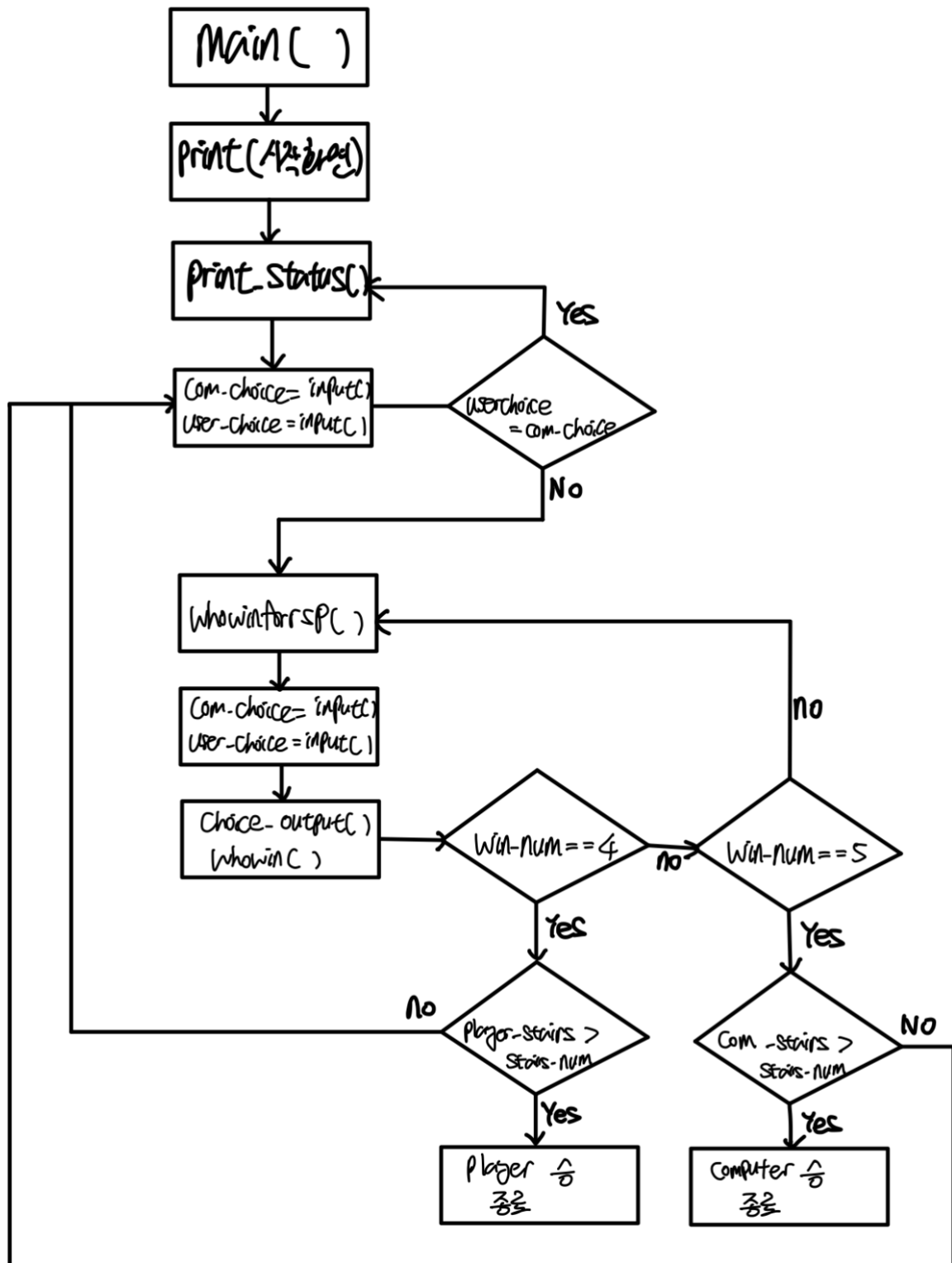
if(win_num == 4):

 USER 승리로 게임 종료, while문 탈출

elif(win_num == 5):

 COMPUTER 승리로 게임 종료, while문 탈출

[순서도]



3. 프로그램 구조 및 설명

a) 계단 개수 입력

- 프로그램을 실행하면 게임 시작 화면을 출력하고 stairs_num 변수에 총 계단의 개수($10 \leq \text{개수} \leq 30$)를 입력받는다. 만약 계단의 개수가 범위를 벗어나면 다시 입력을 받고 범위에 해당한다면 터미널 창을 지우고 현 계단 상태를 출력한다.

b) 공격권 결정 가위바위보

- com_choice 변수에 computer_choice() 함수 호출, COMPUTER의 선택을 대입한다.
- user_choice 변수에 user_select() 함수 호출, USER의 선택을 대입한다.
- user_choice, com_choice를 매개변수로 사용하는 choice_output()을 호출하여 그림을 출력한다.
- com_choice와 user_choice가 같다면 "무승부"를 출력하고 앞의 과정을 다시 반복한다. 만약 같지 않다면 whowinforrsp()를 호출하여 전역변수 win_num을 1 또는 2로 변경하고 공격권 결정 가위바위보 while문을 종료한다.

c) 묵찌빠

- move_num 변수를 이용해 승리 시 이동 칸 수를 출력한다.
- user_choice, com_choice를 매개변수로 사용하는 whowinforrsp()를 호출하여 현재 묵찌빠 공격권을 출력하고 win_num을 변경한다.
- com_choice 변수에 computer_choice() 함수 호출, COMPUTER의 선택을 대입한다.
- user_choice 변수에 user_select() 함수 호출, USER의 선택을 대입한다.
- user_choice, com_choice를 매개변수로 사용하는 choice_output()을 호출하여 그림을 출력한다.
- user_choice, com_choice를 매개변수로 사용하는 whowin()을 호출한다.
- win_num가 4 또는 5가 되면 USER와 COMPUTER의 계단 이동 칸 수를 move_num만큼 증가시키고 만약 이동한 칸 수가 전체 계단의 수보다 많다면 누가 승리했는지를 출력하고 프로그램을 종료한다. 만약 이동한 칸 수가 전체 계단의 수보다 적다면, 즉 아직 계단의 끝에 도달하지 않았다면 이동 칸 수, move_num을 1 증가시키고 c)묵찌빠의 처음으로 돌아간다.

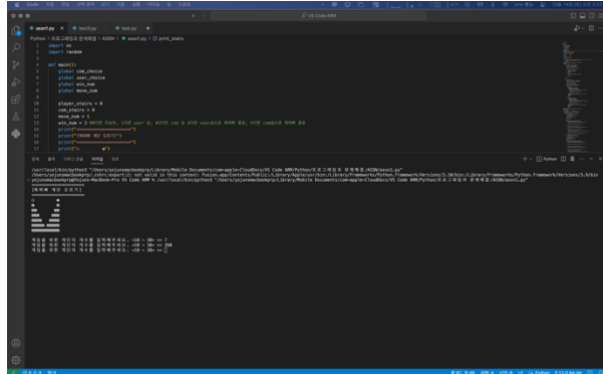
c-1) whowin() 내부

- win_num = 1이라면 USER가 공격권을 지닌다. 이때 user_choice와 com_choice와 같다면 묵찌빠가 종료되었음을 출력하고 win_num을 4로 변경한다. 만약 user_choce와 com_choice가 같지 않다면 whowinforrsp를 호출하여 승패에 따른 win_num을 변경하고 move_num을 1 증가시킨다
- win_num = 2이라면 COMPUTER가 공격권을 지닌다. 이때 user_choice와 com_choice와 같다면 묵찌빠가 종료되었음을 출력하고 win_num을 5로 변경한다. 만약 user_choce와 com_choice가 같지 않다면 whowinforrsp를 호출하여 승패에 따른 win_num을 변경하고 move_num을 1 증가시킨다

4. 프로그램 실행방법 및 예제

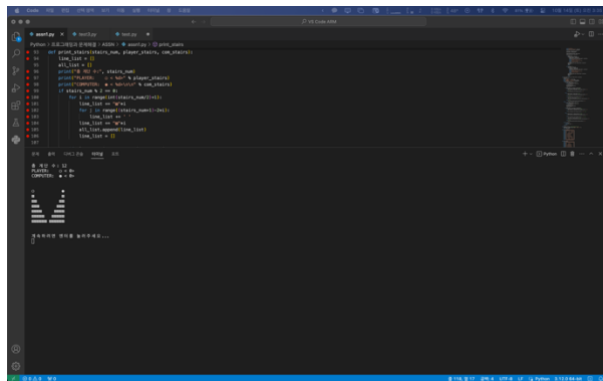
[1. 게임 초기 화면]

입력받는 계단의 개수를 10개~30개로 한정하고 범위를 벗어난 입력에 대해서 다시 입력받는다.

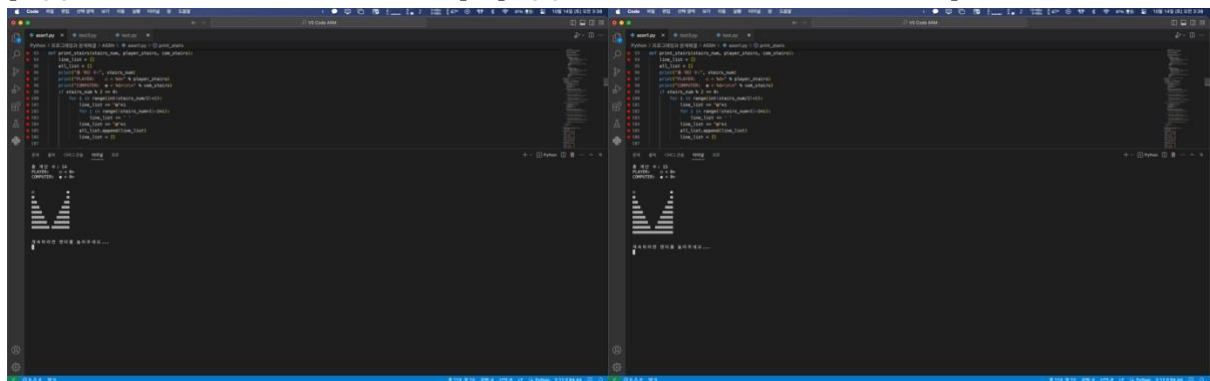


[2. 계단이 생성된 화면 예시(12개의 계단을 입력한 경우)]

범위 내의 계단의 개수를 입력하면 화면을 지운 후 컴퓨터와 플레이를 할 화면이 출력되도록 한다.

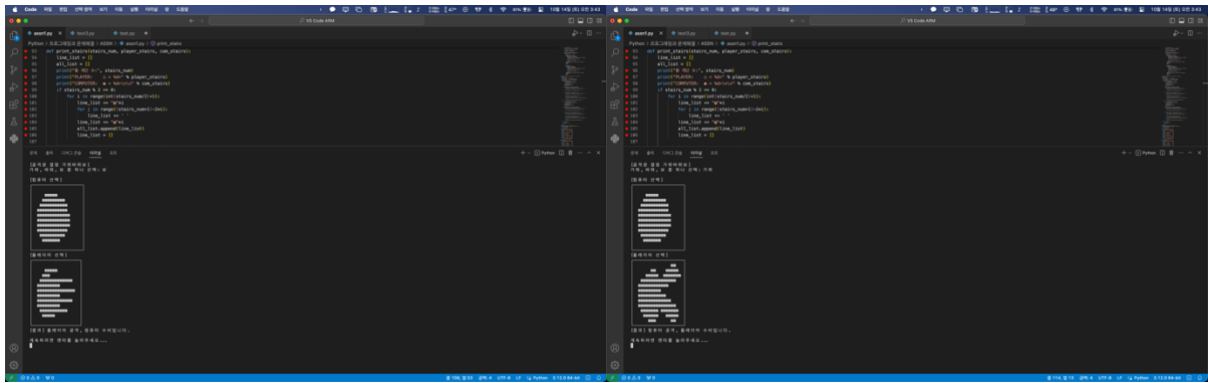


[3.(a) 총 계단의 개수가 홀수인 경우] [3.(b) 총 계단의 개수가 짝수인 경우]



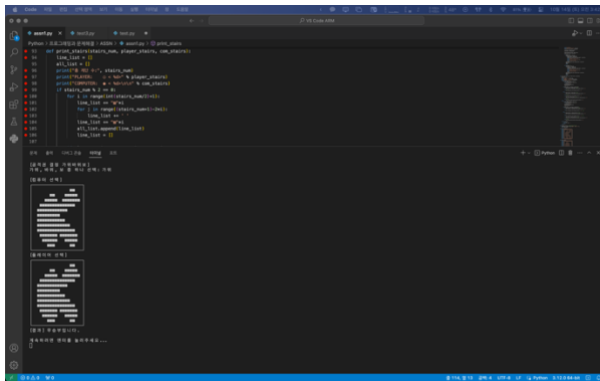
[4.(a) 공격권 결정 가위바위보(플레이어 승)] [4.(a) 공격권 결정 가위바위보(컴퓨터 승)]

플레이어는 '가위', '바위', '보' 중 하나를 선택하여 입력한다. 컴퓨터는 random 모듈의 함수를 사용해 '가위', '바위', '보' 중 하나를 선택하여 입력한다. 가위바위보에서 승부가 가려진 경우 "[결과] 플레이어 공격, 컴퓨터 수비입니다." 또는 "[결과] 컴퓨터 공격, 플레이어 수비입니다."를 출력하고 엔터 입력을 기다린다.



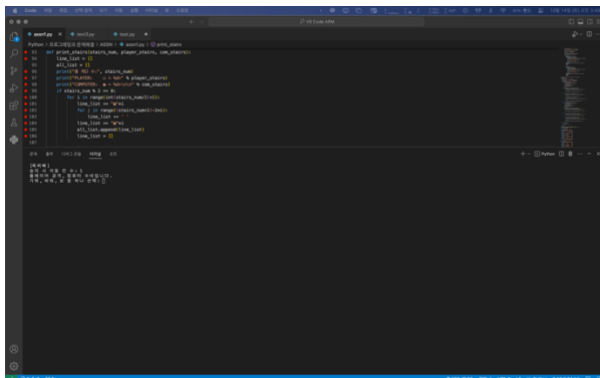
[5. 공격권 결정 가위바위보(무승부)]

공격권 결정 가위바위보에서 승부가 가려지지 못한 경우 "[결과] 무승부입니다."를 출력한 후 엔터 입력을 기다린다. 엔터 입력이 들어온 후 화면을 지우고 위의 과정을 승부가 결정될 때까지 반복한다.



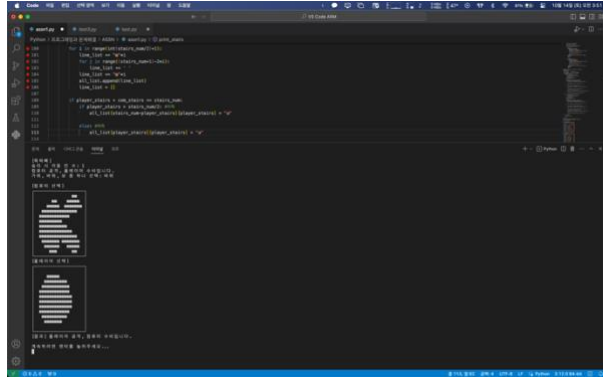
[6. 공격/수비 결정 후 화면 입력 대기]

공격과 수비가 결정된 상태에서 엔터 입력이 들어온 경우 화면을 지우고 다시 가위바위보 입력을 받는다. 화면 최상단에 승리 시 이동 칸 수를 출력하도록 하고 승리 시 이동 칸 수는 처음에 1로 시작하며, 승부가 나지 않을 때마다 1씩 증가한다.



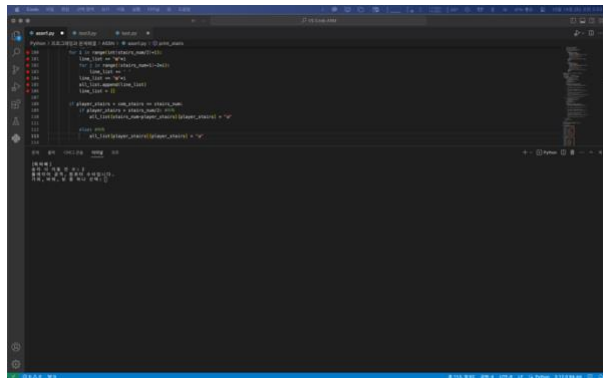
[7. 컴퓨터 공격 실패 후 플레이어 공격권인 상태]

플레이어의 선택 후 승부가 갈리지 않았다면 현 상태에서 묵찌빠 규칙 상 이긴 자가 공격권을 가져가고 승부가 날 때까지 위의 과정을 반복한다.



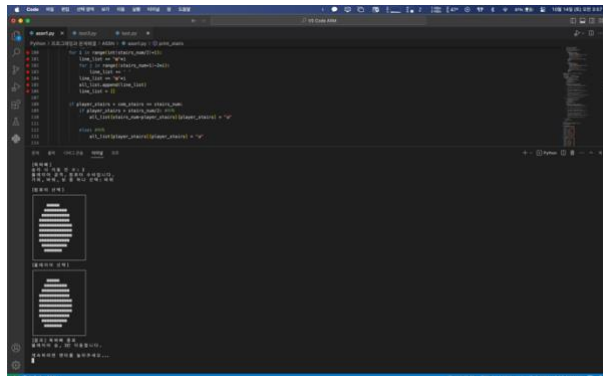
[8. 7의 상태에서 엔터를 입력한 후 화면 대기]

승부가 결정나지 않으면 승부가 날 때까지 위의 과정을 반복하되, 매 턴마다 '승리 시 이동 칸 수'를 1씩 증가시킨다. 엔터를 입력하면 화면을 지우고 다시 가위바위보 입력을 기다린다.



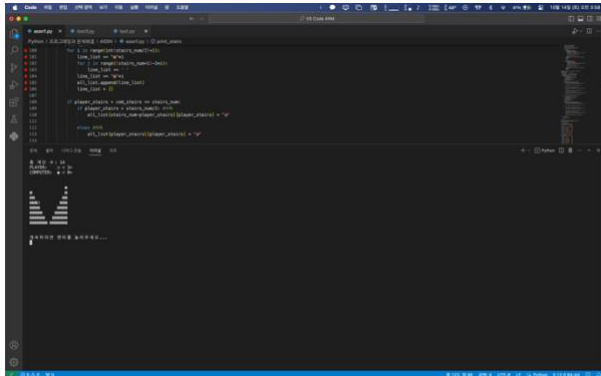
[9. 플레이어의 승리로 승부가 결정되었을 때]

컴퓨터와 플레이어가 같은 선택을 하여 승부가 결정되었을 때 묵찌빠를 종료한다.

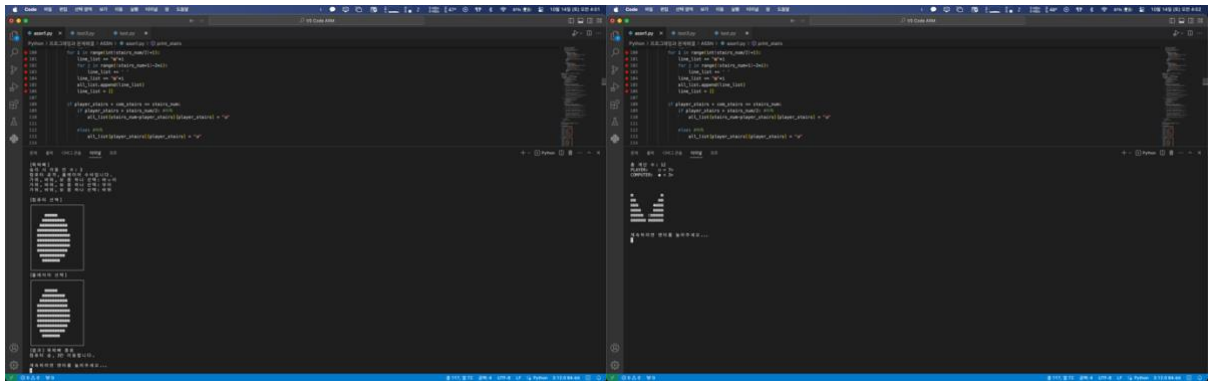


[10. 9와 같이 플레이어가 승리 후 3칸 이동하였을 때]

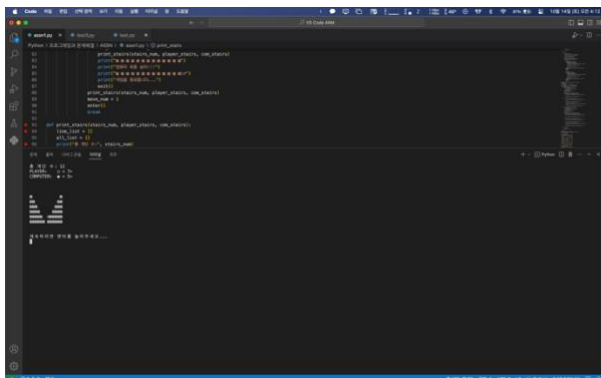
승부가 결정된 상태에서 플레이어가 엔터를 입력하면 화면 지우기 후 바로 다음판을 진행한다. 이때 플레이어와 컴퓨터의 현재 계단에서의 위치를 숫자와 그림으로 표시한다.



[11. 컴퓨터의 승리로 승부가 결정되었을 때] [12. 컴퓨터 승리 후 3칸 이동하였을 때]

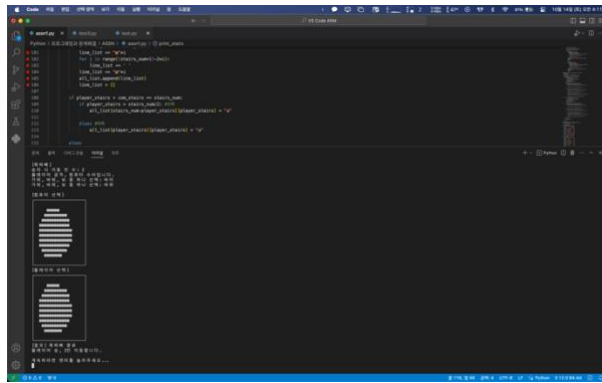


[13. 게임 중간에 플레이어와 컴퓨터의 위치가 같은 경우]



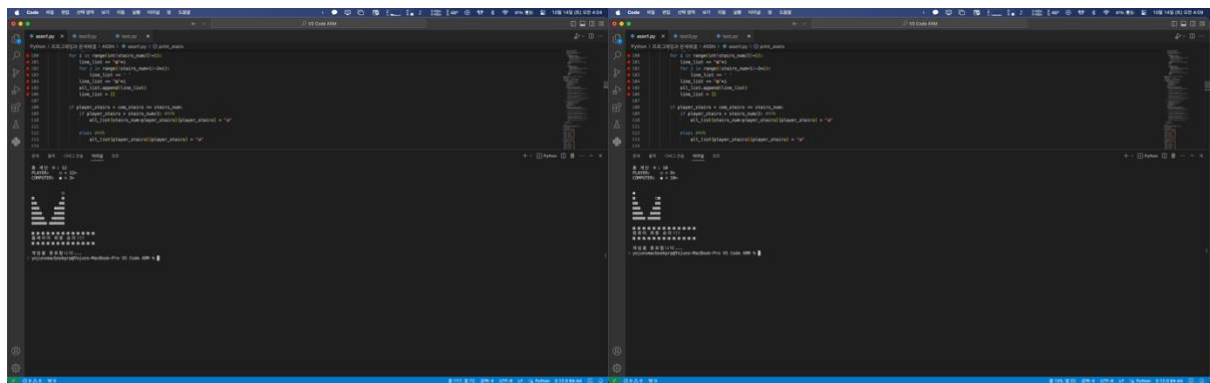
[14. 플레이어의 잘못된 입력 예시]

플레이어가 가위, 바위, 보에서 벗어난 후보를 선택하여 입력하면 특별한 에러메시지 없이 정상적인 입력을 받을 때까지 다시 입력을 받는다.

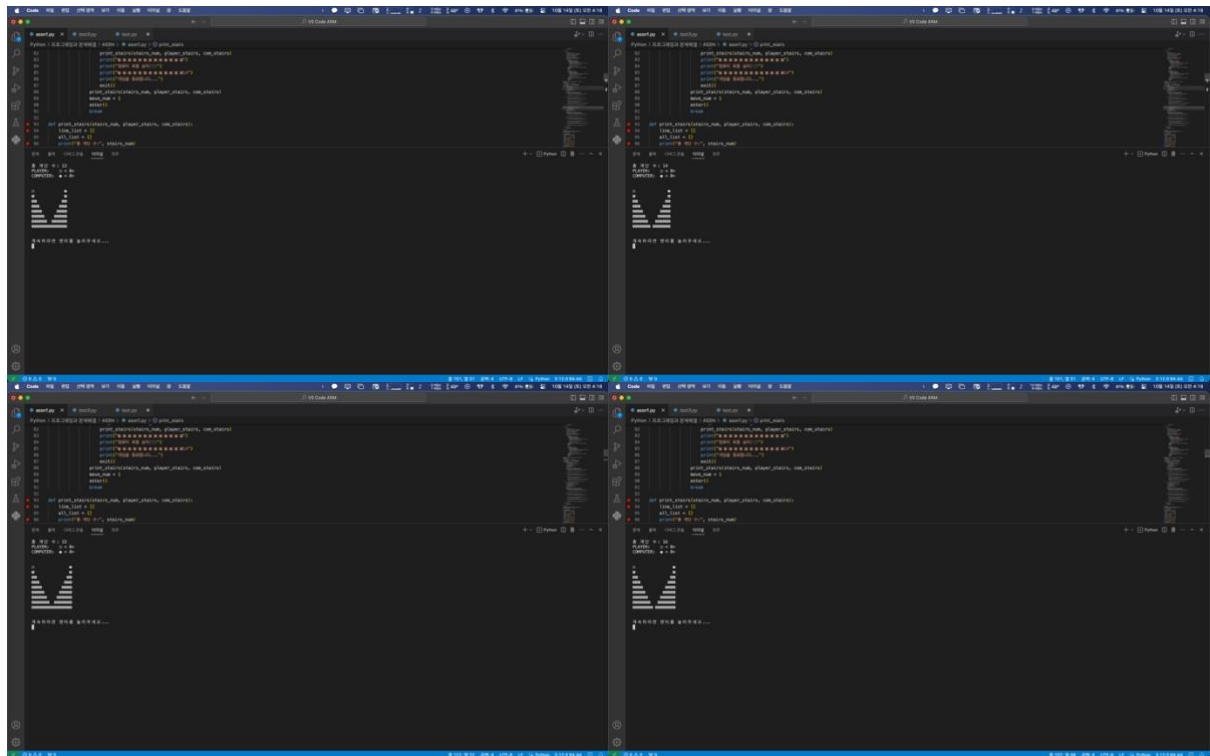


[15. 게임 종료 예시(플레이어 승)] [16. 게임 종료 예시(컴퓨터 승)]

컴퓨터 또는 플레이어가 상대방의 시작점까지 도달한 경우 게임이 끝난다.



[17. 계단 개수 예시]



5. 토론

- print_stairs()함수를 만들 때 2차원 리스트로 만드는 것이 아닌 계단과 플레이어, 컴퓨터 말을 한번에 문자열 형태로 표시하려는 시도를 하다가 많은 시간을 쓰고 실패했다. 2차원 리스트로 계단을 우선 그린 뒤 계단 위치를 인덱싱하여 플레이어, 컴퓨터 말을 배치시키는 방법으로 문제를 해결했다.

6. 결론

- 본 과제에서 프로그램 제작할 때 익숙하지 않았던 2차원 리스트의 제작, 사용방법을 숙지할 수 있었고 여러 자료형과 함수 사용법을 익혔다. 특히 함수에서 매개변수를 주는 것과 반환값을 받는 것을 능숙하게 수행할 수 있었다.

7. 개선방향

- 플레이어와 컴퓨터 말이 겹칠 때 다른 기호로 표시하는 코드가 상당히 긴데 리스트의 다른 기능을 활용하면 좀 더 효율적으로 제작할 수 있을 것이라 기대한다.