

2023년 2학기

프로그래밍과 문제해결

Assignment #3

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.”

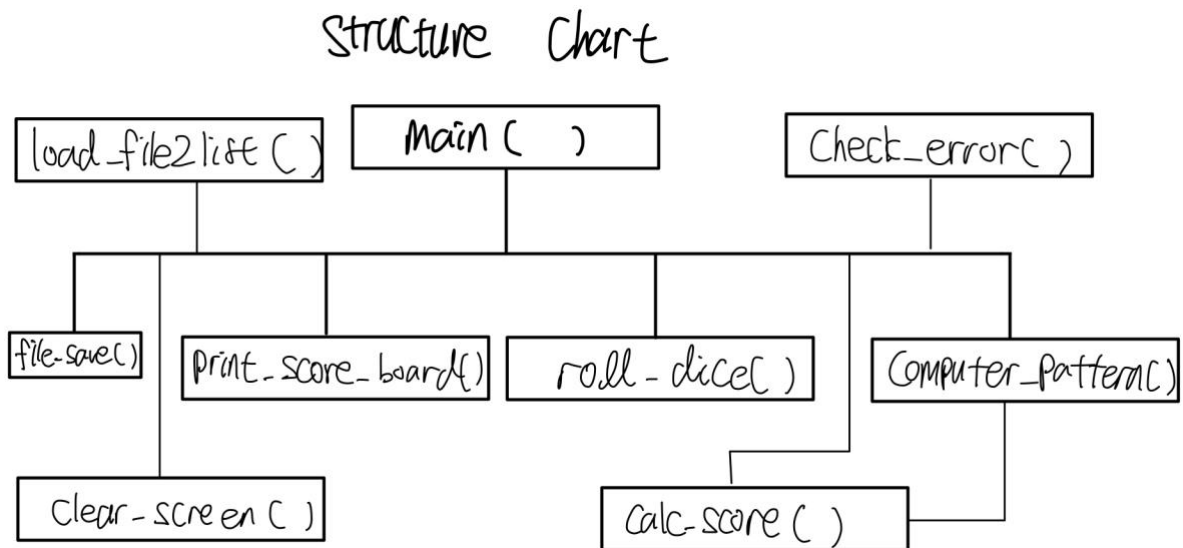
Problem: 포스몬 배틀 (Posmon Battle)

1. 문제의 개요

본 프로그램을 간략시 설명하면 다음과 같다.

- 클래스와 OOP를 활용해 포켓몬스터의 배틀 요소를 간단히 한 포스몬 배틀을 만드는 것이다.

[Structure Chart]

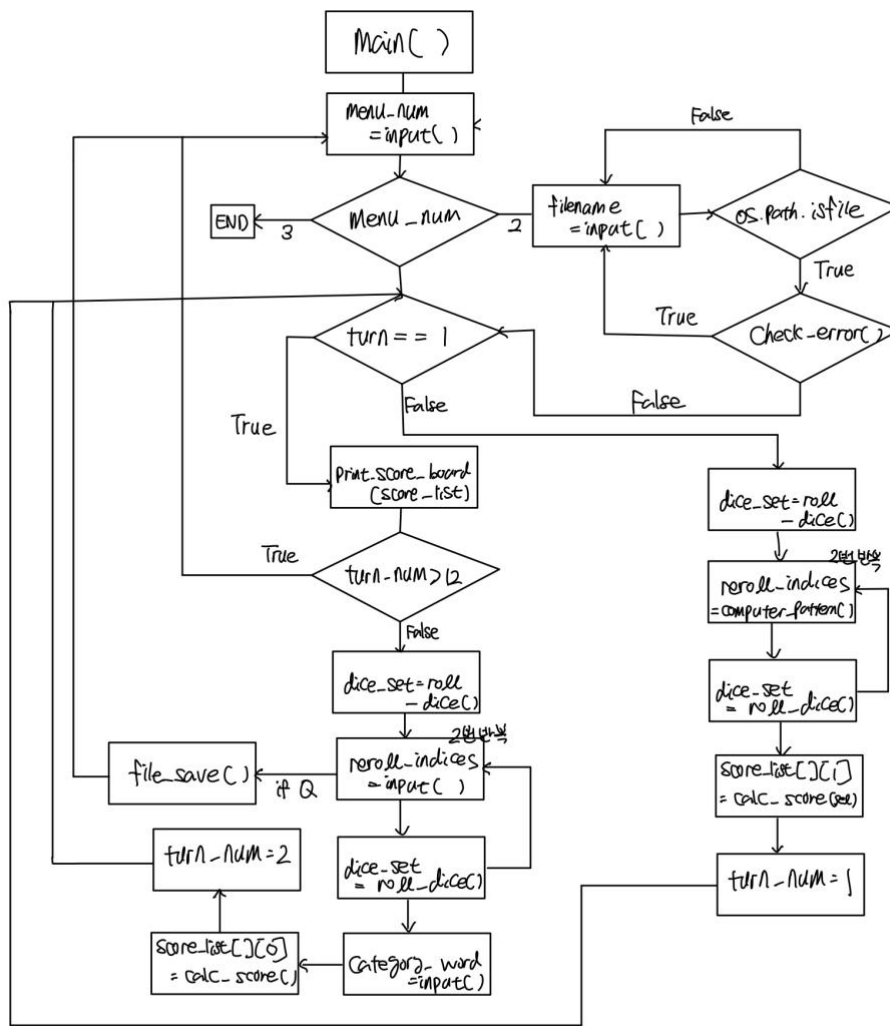


함수명	기능
posmontoO(first_posmon_index_list, second_posmon_index_list)	사용자와 컴퓨터가 현재 가지고 있는 포스몬의 쓰러짐을 OXX, OOO 등의 O/X로 표현해 문자열로 반환하는 함수이다. 코드 내에서 포스몬이 쓰러지면 포스몬을 보관하는 리스트인 posmon_list에서 remove 처리되기 때문에 deepcopy 함수를 사용해 최초의 posmon_list 내에 포함된 posmon의 인덱스 숫자를 저장한 first_posmon_index_list, 그 다음에 일부의 posmon이 쓰러지고 난 후에 남은 posmon의 인덱스 숫자를 담은 second_posmon_index_list를 매개변수로 받고 "XXX", "XX", "X" 문자열에서 살아남은 포스몬의 위치만 O로 바꿔 문자열로 반환하는 함수이다.

auto_change(battle_posmon, first_posmon_index_list, second_posmon_index_list, posmon_list, posmon_name_list, dead_posmon_list, player	포스몬이 쓰러지면 자동으로 다음 포스몬으로 교체해주는 함수이다. Computer 또는 User의 배틀 포스몬 (battle_posmon), first_posmon_index_list, second_posmon_index_list, posmon_list, posmon_name_list, dead_posmon_list, player("user" 또는 "com")을 매개변수로 입력받아 second_posmon_index_list에서 해당 쓰러진 포스몬의 인덱스 번호를 삭제하고 posmon_list에서 그 포스몬을 삭제하고 쓰러진 포스몬을 기록하는(e를 눌렀을 때 죽은 포스몬까지 출력하기 위해)dead_posmon_list에 해당 포스몬을 추가한다. 만약 posmon_list이 이미 공백 리스트 라면(이미 포스몬이 다 죽었다면) 함수를 끝내고 player 매개변수가 "com", "user"이냐에 따라서 "컴퓨터", "당신의"로 출력문이 달라지게 하고 battle_posmon을 다음 포스몬으로 변경한다.
---	---

2. 알고리즘

[flowchart]



3. 프로그램 구조 및 설명

a) 메뉴 번호 입력

- 프로그램을 실행하면 사용자 메뉴 번호 선택화면을 출력하고 menu_num에 0, 1, 2 중 하나를 입력한다. 만약 이 숫자 이외의 입력에 대해서는(숫자가 아닌 문자열 포함)에 대해서는 "잘못된 입력입니다. 다시 입력하세요." 메시지 출력 후 다시 입력받는다. 제대로 된 입력에 대해서는 문자열 숫자를 정수형으로 변환한 후 menu_num에 대입한다.

b) 포스몬 선택(초기 메뉴에서 0을 선택 시)

1. 사용자가 0을 선택하면 배틀에서 사용할 사용자의 포스몬을 선택할 수 있는 화면으로 이동한다. 선택 가능한 포스몬 목록을 출력하고 사용자의 선택을 기다린다. 이때 누른 번호에 따라 각각의 포스몬을 클래스 생성자로 생성하여 사용자의 포스몬을 보관하는 리스트인 user_posmon_list에 해당 포스몬을 추가한다. 이때 user_posmon_list에 포스몬이 추가되는 순간 선택지에 그만두기 항목이 추가되고 posmon_list가 비어있을 때 -1을 입력하면 "잘못된 입력입니다. 다시 입력하세요."

요.”을 출력하고 숫자를 다시 입력받는다. 0~4 이외의 숫자에 관해서도 해당 메시지를 출력하고 다시 입력받는다.

c) 포스몬 배틀(초기 메뉴에서 1을 선택 시)

- 초기 메뉴에서 1을 선택할 때 user_posmon_list가 비어있다면 menu_num을 다시 입력받는다. 만약 비어있지 않다면 컴퓨터의 포스몬을 보관하는 com_posmon_list에 랜덤으로 뽑은 포스몬을 클래스 생성자로 생성하여 추가한다. 같은 포스몬이 추가되지 않도록 겹치지 않게 한다.
- user의 포스몬 목록을 출력하고 com의 포스몬 목록을 출력한 뒤 게임을 시작한다.
- 배틀이 반복되는 반복문 내에서 com의 배틀 포스몬과 user의 배틀 포스몬 상태를 출력하고 user 배틀 포스몬의 스킬을 출력한다.
- user 포스몬 리스트 또는 com 포스몬 리스트가 비어있다면 반복문을 종료하고 승자를 출력한다.
- 사용자 입력을 받는 반복문 내에서 입력이 e이면 user_posmon_list에 있는 포스몬과 user_dead_posmon_list에 있는 포스몬을 모두 출력한다. 입력이 o이고 뒤의 숫자가 스킬 번호에 포함되어 있다면 반복문을 탈출한다. 입력이 s이고 바꿀 포스몬이 현재 user의 포스몬 리스트에 포함되어 있고 현재 배틀 포스몬과 같지 않다면 반복문을 탈출한다. 이외의 입력에 대해서는 “선택할 수 없는 기술입니다” 또는 “포켓몬을 교대시킬 수 없습니다”를 출력하고 명령어를 다시 입력받는다.

c-1) 포스몬 배틀에서 o를 입력 시

- 랜덤 모듈을 사용해 컴퓨터가 사용할 스킬인 com_posmon_skill을 com_battle_posmon.moves에서 고르고 사용자가 입력한 번호에 맞춰서 사용자 포스몬이 사용할 스킬인 user_posmon_skill에 해당 스킬을 대입한다.
- user_posmon_skill의 속도가 com_posmon_skill의 속도보다 빠르다면 user_posmon_skill.use() 함수를 통해 user가 먼저 공격을 시전하고 이때 com_battle_posmon의 체력이 0이하로 떨어지면 auto_change()함수를 통해 com_battle_posmon의 자동 교체를 실행한다. 만약 체력이 0아래로 떨어지지 않는다면 com_posmon_skill.use()함수를 통해 computer가 공격을 시전하고 이때 user_battle_posmon의 체력이 0아래로 떨어지면 auto_change함수를 통해 user_battle_posmon의 자동교체를 실행하고 체력이 0아래로 떨어지지 않는다면 생략한다.
- com_posmon_skill의 속도가 user_posmon_skill의 속도보다 빠르다면 위의 과정의 순서를 반대로 바꾼다.

c-2) 포스몬 배틀에서 s를 입력 시

- user_battle_posmon을 user_posmon_list[]에 사용자가 입력한 숫자의 인덱스를 입력함으로서 바꾸고 user_battle_posmon.reset_status()함수를 통해 해당 포스몬의 능력치를 초기화한다.
- 이후 com이 com_posmon_skill() 함수를 통해 스킬을 시전하고 이때 user_battle_posmon의 체력이 0이하로 떨어지면 auto_change() 함수를 통해 user_posmon의 포스몬의 자동 교체가 이루어지도록 한다.

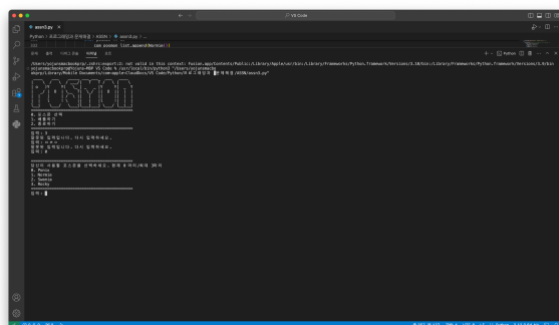
4. 프로그램 실행방법 및 예제

[0. 프로그램 초기 메뉴 화면]

[1. New Game(1을 선택 시)]

- 프로그램을 실행하면 타이틀 화면과 함께 3개의 선택사항이 있는 메뉴를 출력하고 사용자로부터 3가지 선택사항 중 하나를 입력받는다.

0, 1, 2 외의 입력에 대해서 "잘못된 입력입니다. 다시 입력하세요." 메시지 출력 후 다시 입력받는다.



5. 토론

- 점수판을 출력할 때 6칸이 다 채워져있지 않으면 공백, 다 채워져있으면 0 또는 "+35"를 출력하는 코드를 작성하는데 많은 시간을 들였는데 try except를 사용해 해결할 수 있었다.

6. 결론

- 본 2차원 리스트 사용방법을 숙지할 수 파일을 불러오고 저장하는 기능을 습득할 수 있었다. 또한 조건문, 반복문, 사용자 정의함수를 능숙하게 사용할 수 있었다.

7. 개선방향

- 컴퓨터가 다시 굴릴 주사위를 선택할 때 현재 상태를 반영하는 것만이 아닌 최대한 높은 점수를 얻기 위해 가능성을 계산하여 좀 더 다양하고 개선된 선택을 할 수 있는 방향으로 프로그램을 발전시킬 수 있을 것 같다는 생각이 들었다.