

2023년 2학기

# 프로그래밍과 문제해결

## Assignment #3

담당교수: 윤은영

학번: 20230673

학과: 무은재학부

이름: 전재영

---

명예서약(Honor code)

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

---

## Problem: 포스몬 배틀(Posmon Battle)

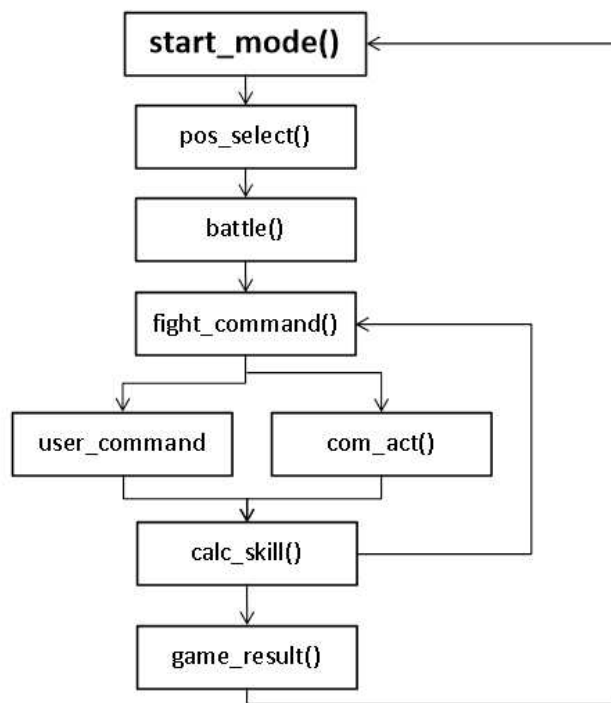
### 1. 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- 포스몬을 최대 3마리까지 선택한다.
- 컴퓨터의 포스몬과 배틀을 진행한다.
- 유저와 컴퓨터가 선택한 기술의 속도를 비교해 계산을 수행한다.
- 중간에 포스몬 교체를 진행할 수 있다. 다른 기술보다 먼저 수행된다.
- 만약 포스몬이 쓰러지면, 다른 포스몬과 교체한다.
- 상대 포스몬을 모두 쓰러뜨리면 승리, 자신의 포스몬이 모두 쓰러지면 패배한다.

### 2. 알고리즘

이 프로그램의 flowchart는 다음과 같다.



이 프로그램의 Pseudo-algorithm 은 다음과 같다.

// 프로그램에 필요한 변수/클래스들은 미리 선언해놓은 것으로 가정한다.

```
start -> get user input
if input == 0 : pos_select()
elif input == 1 : if pos_mon exist -> battle()
```

```

elif input == 2 : exit()

pos_select():
    myposmon[]
    for 3 :
        if user input == 0 : append Ponix(Posmon) to myposmon
        elif user input == 1 : append Normie(Posmon) to myposmon
        elif user input == 2 : append Rocky(Posmon) to myposmon
        elif user input == 3 : append Swania(Posmon) to myposmon

    if posmon exist, user input == -1 : return myposmonp[]

    return myposmon

battle() :
    while game end :
        if user input == e : print all my posmon's status
        elif user input == s, num :
            use skill Change
            use com_act()
        elif user input == o, num :
            compare user's skill and com's skill
            if user is fast -> user skill #, use com_act()
            if com is fast -> use com_act(), user skill #.

    if is_user_win == True : print("승리") -> return to first
    if is_com_win == True : print("패배") -> return to first

```

### 3. 프로그램 구조 및 설명

먼저 프로그램을 이루는 클래스들과 변수들, 함수들을 설명하고, 구조를 설명하겠습니다.

#### 클래스 :

Posmon : 포스몬에 대한 클래스로, health, max\_health, attack, defence, move, name에 대한 변수가 존재한다.

-> get\_##() : 이름/최대 체력/현재 체력/타입/사용할 수 있는 기술 등을 반환한다.

-> reset\_status() : 교체 했을 때, 원래 공격력과 방어력으로 설정을 되돌린다.

Ponix(Posmon), Normie(Posmon), Rocky(Posmon), Swania(Posmon) : Posmon의 하위 클래스로, 각각의 정보로 init하고, get\_type()을 오버라이드 해서 사용할 수 있다.

Move : 사용하는 기술에 대한 클래스이다.

PhysicalMove(Move) : 물리적 공격에 대한 클래스이다.

=> Tackle(PhysicalMove), ScissorsCross(PhysicalMove), Swift(PhysicalMove)

StatusMove(Move) : 변화기에 대한 클래스

=> Growl(StatusMove), SwordDance(StatusMove), TailWhip(StatusMove)

-> use(our, oppo) : 기술을 사용하는 함수, 자신의 포스몬과 적의 포스몬 정보를 이용해 연산을 진행함.

-> get\_speed() : 기술의 스피드를 리턴함.

Change() : 교체에 대한 클래스. 교체도 기술의 일부로 생각해, 호환이 가능하게 함.

#### 변수 :

myposmon, composmon : Posmon 클래스를 담는 리스트입니다. 포스몬을 선택하면, 이 함수에 append하고, 배틀에서 이 리스트에 인덱스로 접근해 클래스를 사용합니다.

is\_my, is\_com : 위의 리스트에서 포스몬이 생존해있는지를 나타내는 리스트입니다. 만약 포스몬이 살아있다면, "O", 아니라면 "X"가 인덱스의 위치에 들어있습니다. 이를 통해 교체 등의 연산을 진행할 수 있습니다.

my\_index, com\_index : 현재 배틀에 나와있는 포스몬의 인덱스 위치를 나타냅니다. 만약 포스몬을 교체하면, 이 값이 변하며, 가리키는 리스트의 인덱스가 바뀌어 가리키는 클래스가 바뀌어 성공적으로 교체할 수 있게 만듭니다.

is\_user\_win, is\_com\_win : 만약 상대의 모두 포스몬을 모두 쓰러뜨렸다면, 유저의 승리, 자신의 포스몬이 모두 쓰러졌다면, 컴퓨터의 승리를 나타냅니다.

#### 함수 :

print\_start(): # 시작 페이지 그림

print\_menu(): # 초기 선택 메뉴

print\_select(i): # 포스몬을 고르는 중에 출력

start\_mode(): # 프로그램에서 모드를 설정합니다.

pos\_select(myposmon, i): # 포스몬을 최대 3개까지 선택

print\_interface(myposmon, composmon): # 배틀을 시작할 때, 포스몬의 항목을 나타내는 함수

battle(myposmon): # 전투를 진행하는 부분입니다.

fight\_command(myposmon, composmon, is\_my, is\_com, my\_index, com\_index): # 전투가 진행되는 메인 함수입니다.

game\_result(is\_user\_win, is\_com\_win): # 게임의 결과를 출력합니다.

com\_replace(is\_com, com\_index): # 컴퓨터의 포스몬이 쓰러졌을 때, 교체하는 과정입니다.

user\_replace(is\_my, my\_index): # 유저의 포스몬이 쓰러졌을 때, 교체하는 과정입니다.

attack\_oper(skill, myposmon, my\_index, toposmon, to\_index): # 기술을 사용하는 함수입니다.

calc\_skill(attack\_name): # 공격 유형을 통해 공격의 클래스와 속도를 리턴

com\_act(composmon, is\_com, com\_index): # 컴퓨터의 행동과정입니다.

print\_status(myposmon, composmon, is\_my, is\_com, my\_index, com\_index): # 포스몬의 정보를 게임 화면과 같이 출력합니다.

ss\_result(our\_type, oppo\_type): # 상성을 통해 공격 배율 계산

game\_result(is\_user\_win, is\_com\_win): # 게임의 결과를 출력합니다.

구조도에 대한 설명은 다음과 같습니다.

#### 1) 프로그램 실행

- 프로그램이 실행되면 print\_start()를 통해 게임 타이틀을 출력하고, start\_mode()를 실행해 유저의 명령을 받습니다.
- 만약 0을 입력한 경우, pos\_select()를 통해 포스몬을 선택합니다.
- 만약 1을 입력한 경우, battle()을 통해 배틀을 진행합니다. 만약 포스몬을 선택하지 않은 경우, 메뉴로 돌아갑니다.
- 만약 2를 입력한 경우, 프로그램을 종료합니다.
- 이 외의 입력이 들어온 경우는 다시 입력을 받습니다. (정수 이외의 입력을 고려하지 않습니다.)

#### 2) 포스몬 선택

- 포스몬을 최대 3마리까지 선택하는 과정입니다. print\_select(i)를 이용해 고를 수 있는 포스몬을 출력합니다. 지금까지 고른 포스몬의 수 i를 이용해 지금까지 고른 포스몬의 수도 나타냅니다.
- 만약 0을 골랐을 경우, Ponix, 1 : Normie, 2 : Swania, 3 : Rocky 의 클래스를 복사해 리스트에 추가합니다.
- 포스몬을 한 마리 이상 고른 경우, 명령어에 -1. 그만두기를 추가해, 중간에 종료할 수 있게 합니다.
- 만약 포스몬을 모두 선택했을 경우 (3마리를 모두 채운 경우), 이를 나타내고, 다시 메뉴로 돌아갑니다.

#### 3) 배틀

- battle()을 통해 배틀 페이지로 진행됩니다. 컴퓨터의 포스몬을 랜덤으로 3개 선택해 리스트에 저장하고, print\_interface()를 이용해 선택된 포스몬을 나타냅니다.
- 모든 포스몬을 선택된 후, fight\_command()를 통해 배틀을 진행합니다.
- print\_status()를 이용해 게임 화면 창을 출력하고, 유저의 입력을 받습니다.
- 만약 e를 입력한 경우 현재 남아있는 자신의 포스몬의 정보를 모두 출력합니다.
- 만약 o를 입력한 경우, 다음에 오는 숫자의 스킬을 사용합니다. 입력한 것이 정수가 맞는지, 스킬 범위에 존재하는지를 판단하고, 컴퓨터의 행동도 com\_act()로 받아 옵니다. 그 후, calc\_skill로 정보를 받아오고, 유저의 행동과 컴퓨터 행동의 속도를 계산해, attack\_oper()로 결과를 수행합니다.
- 만약 도중에 자신이나 컴퓨터의 포스몬이 쓰러진 경우, 선택한 스킬을 진행하지 않고, 남은 포켓몬 중에서 교체를 진행합니다.
- 만약 s를 입력한 경우, 교체를 진행합니다. 뒤에 오는 숫자의 포켓몬과 교체를 진행하며, 만약 불가능한 경우, 다시 입력을 받습니다. 교체도 스킬의 일종으로 구현해 위의 공격 과정과 동일하게 진행합니다. 속도를 100으로 설정해, 다른 모든 스킬보다 먼저 진행합니다.

#### 4) 게임 종료

- 만약 상대 포스몬을 모두 쓰러뜨린 경우 is\_user\_win를 True로 만들고, game\_result()로 승리를 출력합니다. 모두 쓰러진 경우, is\_com\_win를 True로 만들고, 패배를 출력합니다.
- 그 후, 다시 처음의 메뉴로 돌아갑니다.

#### 4. 프로그램 실행방법 및 예제

- vscode를 이용해 제작과 실행을 했습니다.

```
=====
0. 포스몬 선택
1. 배틀하기
2. 종료하기
=====
입력: 4
잘못된 입력입니다. 다시 입력하세요
=====
0. 포스몬 선택
1. 배틀하기
2. 종료하기
=====
입력: []
```

1. 프로그램을 실행하면, 다음과 같이 타이틀을 출력하고, 입력을 받습니다. 만약 범위를 벗어  
나는 입력의 경우, 다시 입력을 받습니다.

```
=====
0. 포스몬 선택
1. 배틀하기
2. 종료하기
=====
입력: 0
=====
당신이 사용할 포스몬을 선택하세요. 현재 0 마리/최대 3 마리
0. Ponix
1. Normie
2. Swania
3. Rocky
=====
입력: 0
=====
당신이 사용할 포스몬을 선택하세요. 현재 1 마리/최대 3 마리
0. Ponix
1. Normie
2. Swania
3. Rocky
-1. 그만두기
=====
입력: 3
```

```

=====
당신이 사용할 포스몬을 선택하세요. 현재 2 마리/최대 3 마리
0. Ponix
1. Normie
2. Swaffia
3. Rocky
-1. 그만두기
=====
입력: 5

잘못된 입력입니다. 다시 입력하세요.

입력: -1

=====
당신의 포스몬 목록: Ponix Rocky
=====

=====
0. 포스몬 선택
1. 배틀하기
2. 종료하기
=====
입력:

```

2. 위의 메인 메뉴에서 0번을 누른 경우 포스몬 선택으로 이동합니다. 최대 3마리의 포스몬을 선택할 수 있습니다.

```

=====
0. 포스몬 선택
1. 배틀하기
2. 종료하기
=====
입력: 1

싸울 포스몬이 없습니다! 먼저 포스몬을 선택해 주세요.

=====
0. 포스몬 선택
1. 배틀하기
2. 종료하기
=====
입력:

```

2-1 만약 포스몬을 고르지 않고, 배틀을 진행하려고 하면, 다시 메뉴에서 선택합니다.

```

=====
0. 포스몬 선택
1. 배틀하기
2. 종료하기
=====
입력: 1

=====
당신의 포스몬 목록 Ponix Normie
컴퓨터 포스몬 목록 Ponix Swania Normie
=====

배틀이 시작됩니다.
#####
컴퓨터 포스몬: [000] 3 / 3
Ponix <|Paper 86 / 86:
vs
Ponix <|Paper 86 / 86:
당신의 포스몬: [00] 2 / 2
+++++
기술 (0) Tackle (1) Growl (2) SwordDance
#####
입력: 

```

3-1. 배틀을 진행하면 다음과 같은 디스플레이를 출력하고, 입력을 기다립니다.

```

배틀이 시작됩니다.
#####
컴퓨터 포스몬: [000] 3 / 3
Swania <|Scissors 80 / 80:
vs
Normie <|Nothing 80 / 80:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Swift (2) TailWhip
#####
입력: o 2
당신의 Normie: TailWhip 기술 사용
- Swania 포스몬의 [방어력] 5 감소 (10->5)
컴퓨터의 Swania: ScissorsCross 기술 사용
- Normie 포스몬의 [체력] 40 감소 (80->40)

#####
컴퓨터 포스몬: [000] 3 / 3
Swania <|Scissors 80 / 80:
vs
Normie <|Nothing 40 / 80:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Swift (2) TailWhip
#####
입력: 

```

3-2. 스킬을 사용한 예.



```
#####

입력: e
남은 포스몬의 숫자 : 3 / 3
(0) Normie    <|Nothing  40 / 80|
(1) Swania    <|Scissors  80 / 80|
(2) Rocky     <|Rock    80 / 80|

#####
컴퓨터 포스몬: [000] 3 / 3
Swania    <|Scissors 80 / 80:
      vs
Normie    <|Nothing 40 / 80:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Swift (2) TailWhip
#####

입력: s 2

당신의 포스몬 Rocky로 교대
컴퓨터의 Swania: SwordDance 기술 사용
- Swania 포스몬의 [공격력] 10 증가 (30->40)
#####
컴퓨터 포스몬: [000] 3 / 3
Swania    <|Scissors 80 / 80:
      vs
Rocky     <|Rock 80 / 80:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Growl
#####

입력: □
```

3-3 교체를 진행할 경우

입력: o 2  
컴퓨터의 Rocky: Growl 기술 사용  
- Ponix 포스몬의 [공격력] 5 감소 (30->25)  
당신의 Ponix: SwordDance 기술 사용  
- Ponix 포스몬의 [공격력] 10 증가 (25->35)

```
#####
컴퓨터 포스몬: [000] 3 / 3
Rocky    <|Rock 80 / 80:
      vs
Ponix    <|Paper 86 / 86:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Growl (2) SwordDance
#####
```

입력: o 2  
당신의 Ponix: SwordDance 기술 사용  
- Ponix 포스몬의 [공격력] 10 증가 (35->45)  
컴퓨터의 Rocky: Tackle 기술 사용  
- Ponix 포스몬의 [체력] 17 감소 (86->69)

```
#####
컴퓨터 포스몬: [000] 3 / 3
Rocky    <|Rock 80 / 80:
      vs
Ponix    <|Paper 69 / 86:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Growl (2) SwordDance
#####
```

입력: s 1

당신의 포스몬 Normie로 교대  
컴퓨터의 Rocky: Growl 기술 사용  
- Normie 포스몬의 [공격력] 5 감소 (20->15)

```
#####
컴퓨터 포스몬: [000] 3 / 3
Rocky    <|Rock 80 / 80:
      vs
Normie   <|Nothing 80 / 80:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Swift (2) TailWhip
#####
```

```

입력: s 0

당신의 포스몬 Ponix로 교대
컴퓨터의 포스몬 Ponix로 교대
#####
컴퓨터 포스몬: [000] 3 / 3
Ponix <|Paper 86 / 86:
vs
Ponix <|Paper 69 / 86:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Growl (2) SwordDance
#####

입력: o 2
당신의 Ponix: SwordDance 기술 사용
- Ponix 포스몬의 [공격력] 10 증가 (20->30)
컴퓨터의 Ponix: SwordDance 기술 사용
- Ponix 포스몬의 [공격력] 10 증가 (20->30)

#####
컴퓨터 포스몬: [000] 3 / 3
Ponix <|Paper 86 / 86:
vs
Ponix <|Paper 69 / 86:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Growl (2) SwordDance
#####

입력: 

```

3-4 포스몬의 능력치가 변화한 후, 다시 필드에 등장한 경우, 능력치가 초기화됩니다.

```

#####
컴퓨터 포스몬: [000] 3 / 3
Swania <|Scissors 35 / 80:
vs
Ponix <|Paper 69 / 86:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Growl (2) SwordDance
#####

입력: o 0
당신의 Ponix: Tackle 기술 사용
- Swania 포스몬의 [체력] 45 감소 (35->-10)
컴퓨터 Swania: 쓰러짐
컴퓨터 포스몬 Rocky로 교대
#####
컴퓨터 포스몬: [X00] 2 / 3
Rocky <|Rock 20 / 80:
vs
Ponix <|Paper 69 / 86:
당신의 포스몬: [000] 3 / 3
+++++
기술 (0) Tackle (1) Growl (2) SwordDance
#####

입력: 

```

3-5 포스몬이 쓰러진 경우, 그 위치의 포스몬을 X로 표시합니다.





## 5. 토론

- 사용한 모듈은 컴퓨터의 의사결정을 위한 random과, 클래스의 deepcopy를 위한 copy모듈을 사용하였다.
- 포스몬이 살아 있는지를 판단할 때, is\_my에 "O"가 있는지를 판단하는 형식으로 구현했다. "O"의 수를 세고, 그 위치를 리턴하는 형식으로 구현해서 약간 돌아가긴 하지만, is\_my를 그대로 출력해서 디스플레이할 수 있게 만들었다.
- 교체를 구현할 때, Change()와 같이 클래스로 구현해서, 다른 Move()에 존재하는 기술들과 같은 방식으로 작동하게 했습니다.

## 6. 결론

- 본 과제를 통해 파이썬에서의 클래스를 만들고 구현하면서, OOP에 대한 개념을 익힐 수 있었다. 클래스를 만들고 상속하며, override에 대한 개념도 익힐 수 있었다. 여러 코드로, 게임으로 구현해 보면서 성취감도 얻을 수 있었던 시간이었던 것 같다.

## 7. 개선방향

- fight\_command() 함수가 매우 비효율적으로 짜였다. 교체를 한 상황과 스킬을 사용한 상황이 같은데, 이를 그냥 함수에 넣어서 코드가 매우 길어졌고, 순서를 판단 할 때도, 함수를 만들지 않아서 코드가 4배로 되었다. 그래서 코드를 수정할 때, 매우 불편했던 기억이 있다.
- is\_my와 같이 구현해서, 출력은 쉽지만, 포켓몬을 교체하는 경우에서 코드가 복잡해진 것 같다.
- 변수의 init이 출력 함수에서 받아오는 부분이 있는 등, 문제가 있다.