

# 2021-1 C++ 프로그래밍 실습과제 07

## (1) 각 문제에 대한 분석과 및 해결 방법

7.1 8절의 Monster World 프로그램을 다음과 같이 확장하라.

- (1) 프로그램 7.11의 Monster클래스에 “에너지”를 저장하는 속성(int nEnergy;)을 추가하라.
- (2) 맨 처음에는 모든 몬스터 객체의 에너지가 100이 되도록 생성자의 멤버 초기화 리스트를 이용해 초기화하라.
- (3) Monster클래스의 eat()함수를 다음과 같이 수정하라.
  - 아이템을 먹으면 에너지가 8 증가됨
  - 아이템을 먹지 못하면 에너지가 1 감소됨. 에너지의 최솟값은 0.
- (4) Monster클래스의 print()함수를 수정하여 다음과 같이 현재의 에너지 레벨이 출력되도록 하라.

[문제분석 및 해결방법] : 8절의 Monster World 프로그램을 확장하여 에너지를 저장하는 멤버 변수를 추가하고 생성자 멤버 초기화 리스트를 이용하여 100으로 초기화하였다. 또한 eat() 함수에서 아이템을 먹거나 먹지 못함에 따라서 에너지의 증감을 처리하는 부분을 추가하여 괴물의 에너지가 집계될 수 있도록 하였다. print() 함수에 현재 에너지 레벨을 추가하는 삽입하여 게임이 완료 된 후 각 몬스터가 얼마만큼의 에너지를 모았는지 확인할 수 있도록 하고 몬스터의 이름을 J,  $\partial$ ,  $\Delta$ ,  $\nabla$  등의 수학 기호로 수정하여 수학 기호 몬스터들이 MonsterWorld에서 활동할 수 있게 코드를 수정하였다.

## (2) 자신이 구현한 주요 코드

7.1 8절의 Monster World 프로그램을 다음과 같이 확장하라.

```
<LJH_07_1.cpp>
#include "MonsterWorld.h"
#include <time.h >
int main()
{
    srand((unsigned int)time(NULL));
    int w =16, h =8;
    MonsterWorld game(w, h);
    Monster m("Integral", "J", rand() % w, rand() % h);
    game.add(m);
    game.add(Monster("PartialD", " $\partial$ ", rand() % w, rand() % h));
    game.add(Monster("Delta", " $\Delta$ ", rand() % w, rand() % h));
    game.add(Monster("Del", " $\nabla$ ", rand() % w, rand() % h));
    game.play(500, 10);
    printf("-----게임 종료-----\n");
    std::cout <<std::endl <<"Press ENTER to exit..."; fflush(stdin); getchar(); getchar();
    return 0;
}
```

```

<Monster.h>
#pragma once
#include "Canvas.h"
#define DIM 40
class Monster {
    string name, icon;
    int x, y, nItem;
    int nEnergy;
    void clip(int maxx, int maxy) {
        if (x < 0) x = 0;
        if (x >= maxx) x = maxx - 1;
        if (y < 0) y = 0;
        if (y >= maxy) y = maxy - 1;
    }
    void eat(int map[DIM][DIM]) {
        if (map[y][x] == 1) {
            map[y][x] = 0;
            nItem++;
            nEnergy += 8;
        }
        else {
            nEnergy > 0 ? nEnergy -- : 0;
        }
    }
public:
    Monster(string n = "나괴물", string i = "※", int px = 0, int py = 0)
        : name(n), icon(i), x(px), y(py), nItem(0), nEnergy(100) {}
    ~Monster() { cout << "Wt" << name << icon << " 물러갑니다~~~~Wn"; }
    void draw(Canvas & canvas) { canvas.draw(x, y, icon); }
    void move(int map[DIM][DIM], int maxx, int maxy) {
        switch (rand() % 8) {
            case 0: y --; break;
            case 1: x ++; y --; break;
            case 2: x ++; break;
            case 3: x ++; y ++; break;
            case 4: y ++; break;
            case 5: x --; y ++; break;
            case 6: x --; break;
            case 7: x --; y --; break;
        }
        clip(maxx, maxy);
        eat(map);
    }
    void print() { cout << "Wt" << name << icon << ":" << nItem << " " << "E:" << nEnergy << endl; }
};

```

### (3) 다양한 입력에 대한 테스트 결과

7.1 8절의 Monster World 프로그램을 다음과 같이 확장하라.

The image shows two screenshots of a Windows command prompt window running the 'Monster World' program. The window title is 'C:\Users\HLee\source\repos\2020136110이진형\_7장\Debug\UH\_07\_1.exe'. The program title is '[ Monster World (수학 기호 몬스터들의 세상) ]'. The first screenshot shows the initial state with a 10x10 grid of black squares. The second screenshot shows the state after a move, with the grid updated and the status bar showing new values.

```
C:\Users\HLee\source\repos\2020136110이진형_7장\Debug\UH_07_1.exe
[ Monster World (수학 기호 몬스터들의 세상) ]

  Δ  f
  ▽
  θ

전체 이동 횟수 = 0
남은 아이템 수 = 128
Integral f : 0 E:100
PartialDθ : 0 E:100
DeltaΔ : 0 E:100
Del▽ : 0 E:100
엔터를 누르세요...
```

```
C:\Users\HLee\source\repos\2020136110이진형_7장\Debug\UH_07_1.exe
[ Monster World (수학 기호 몬스터들의 세상) ]

  f  θ
  ▽
  Δ  ▽
  θ

전체 이동 횟수 = 30
남은 아이템 수 = 63
Integral f : 16 E:214
PartialDθ : 21 E:259
DeltaΔ : 10 E:160
Del▽ : 18 E:232
```



## (4) 코드에 대한 설명 및 해당 문제에 대한 고찰

7.1 8절의 Monster World 프로그램을 다음과 같이 확장하라.

- (1) 프로그램 7.11의 Monster클래스에 “에너지”를 저장하는 속성(int nEnergy;)을 추가하라.

```
<Monster.h>
#pragma once
#include "Canvas.h"
#define DIM 40
class Monster {
    ~ ... ~
    int nEnergy;
    ~ ... ~
};
```

클래스 멤버 변수로 정수형 변수인 nEnergy를 추가하였다.

- (2) 맨 처음에는 모든 몬스터 객체의 에너지가 100이 되도록 생성자의 멤버 초기화 리스트를 이용해 초기화하라.

```
<Monster.h>
~ ... ~
public:
    Monster(string n ="나괴물", string i ="*", int px =0, int py =0)
        : name(n), icon(i), x(px), y(py), nitem(0), nEnergy(100) {}
    ~ ... ~
};
```

생성자 멤버 초기화 리스트를 활용하여 nEnergy 변수를 100으로 초기화하였다.

- (3) Monster클래스의 eat()함수를 다음과 같이 수정하라.

- 아이템을 먹으면 에너지가 8 증가됨
- 아이템을 먹지 못하면 에너지가 1 감소됨. 에너지의 최솟값은 0.

```
<Monster.h>
~ ... ~
class Monster {
    ~ ... ~
    void eat(int map[DIM][DIM]) {
        if (map[y][x] ==1) {
            map[y][x] =0;
            nitem++;
            nEnergy +=8;
        }
        else {
            nEnergy >0 ? nEnergy -- : 0;
        }
    }
    ~ ... ~
};
```

eat 함수를 수정하여 아이템을 먹거나 먹지 못하는 경우에 따라 에너지를 획득하고 감소하도록 조건문을 추가하였다.

- (4) Monster 클래스의 print() 함수를 수정하여 다음과 같이 현재의 에너지 레벨이 출력되도록 하라.

```
<Monster.h>
~ ... ~
class Monster {
    ~ ... ~
    void print() { cout <<"Wt"<< name << icon <<":"<< nitem <<" "<<"E:"<< nEnergy <<endl; }
};
```

print 함수를 수정하여 현재 에너지 레벨을 nEnergy 변수의 값을 통해 출력하는 부분을 추가하였다.

## (5) 이번 과제에 대한 느낀점

이번 과제를 해결하기 위해 교재의 코드를 분석하고 조사하면서 모르고 있었던 세부적인 내용을 많이 알 수 있었다. 특히 생성자 멤버 초기화 리스트의 다양한 용도(상수 멤버 초기화, 참조자 변수 초기화 등)에 대해 알아볼 수 있어서 좋았다.