

2021-1 C++ 프로그래밍 실습과제 08

학번	2020136129	이름	최수연
----	------------	----	-----

(1) 각 문제에 대한 분석과 및 해결 방법

3. 8.8절의 Monster World 프로그램을 다음과 같이 확장하라.

(0) 먼저 8.8절의 Monster World 프로그램을 구현하고 동작을 확인하라.

(1) 이 프로그램의 Monster클래스에 실습문제 7.3과 같이 “에너지”를 나타내는 속성을 추가하고, 관련 함수들을 수정하라. Monster의 생성자와 eat(), print()함수를 수정하면 된다.

(2) Monster클래스에 “에너지” 값을 반환하는 getEnergy()함수를 추가하라. 이 함수는 private 멤버인 nEnergy를 반환한다.

(3) 이제 에너지가 0이 되면 몬스터가 사라지도록 프로그램을 수정하라. 이를 위해 MonsterWorld클래스에 checkStarvation()함수를 추가한다. 이 함수에서는 모든 몬스터의 에너지 레벨을 검사하여 0인 몬스터들을 모두 동적으로 해제한다. 특히 pMon 배열에서 삭제할 항목의 처리에 유의해야 한다. 힌트: k번째 항목을 삭제하는 경우 맨 마지막 항목을 k번째 항목에 복사하고 몬스터의 수를 1 감소하면 된다.

(4) Monster클래스에 몬스터의 수를 나타내는 정적 멤버 변수를 추가하라. 또한 생성자와 소멸자에서 이 멤버 변수의 값을 적절히 갱신하고, 현재 몬스터의 수를 출력하는 정적 멤버 함수 printCount()를 구현하라. 화면을 갱신할 때 마다 이 함수를 호출하여 전체 몬스터의 수를 출력한다. (이 정적 변수의 값은 MonsterWorld의 nMon과 동일하도록 유지되어야 한다.)

(5) 화면 출력은 다음의 예를 참고하여 구성하라.

[문제분석 및 해결방법]

(0) 8.8절 Monster World 프로그램이 잘 동작되는 것을 확인하였다. 아래 문제는 본서의 8.7절 Matrix 클래스와 8.8절 Monster World 프로그램 코드를 전제로 하였다.

(1) 지난 실습 과제7에서와 같이 동일한 방식으로 코드 추가 및 수정하였다.

(2) Monster클래스의 public에 getEnergy()함수를 int형으로 선언하여, private에 있는 int형 nEnergy를 반환할 수 있도록 하였다. 이를 통해 private으로 선언된 멤버 변수도 public을 통해 꺼내어 다른 파일에서 사용할 수 있게 된다.

(3) MonsterWorld클래스의 private에 void로 checkStarvation()함수를 추가하였다. checkStarvation()함수 안에서 for문을 통해 몬스터의 수만큼 반복하고 if문을 통해 pMon 배열의 한 몬스터가 가리키는 에너지, 즉 해당 몬스터의 에너지가 0이 되면 0인 에너지를 갖는 몬스터 객체의 해당 배열을 동적으로 해제하여 없앴다. 그 후 배열의 마지막 항목, 'nMon -1'번째에 있는 pMon[nMon -1]을 동적으로 해제한 몬스터 배열 위치에 옮긴다. 그리고 코드 nMon--를 추가하여 몬스터의 수를 1 감소시킨다. 다음 몬스터가 감소하였음을 알리는 문자를 출력한다. 출력 후에는 화면을 한번 멈춰 보여주기 위해 getchar()을 사용한다. 이때 checkStarvation()함수는 MonsterWorld클래스의 play()함수에서 호출하여 사용한다.

(4) Monster클래스의 public에 static을 사용하여 int형 정적 멤버 변수 nMonster을 선언해주었다. 이때 cpp 파일에서 nMonster를 0으로 초기화하였다. Monster의 생성자에서 생성자가 실행될 때 nMonster의 값을 1씩 더해주고, Monster의 소멸자가 실행될 때 nMonster를 1씩 감소하여 nMonster 멤버 변수의 값을 갱신한다. Monster클래스의 public에 static으로 void형 정적 멤버 함수 printCount()를 선언하여 남은 몬스터의 수를 화면에 출력하도록 한다. 이때 printCount()는 MonsterWorld클래스의 play()함수에서 호출한다.

(2) 자신이 구현한 주요 코드

Monster.h	
#define INIT_ENERGY 30 #define ITEM_ENERGY 5	매크로 함수는 몬스터 에너지 기본 값을 30, 아이템(지방)의 에너지 값을 5로 두었다.
class Monster { ... int nEnergy; ... void eat(int map[DIM][DIM]) { if (map[y][x] == 1) { map[y][x] = 0; nItem++; nEnergy += ITEM_ENERGY; } else if (nEnergy > 0) nEnergy--; } ... }	Monster 클래스 내에 int형 에너지 멤버 변수 nEnergy를 추가하였다. 아이템(지방)을 먹으면 기존 에너지 값에 5가 더해지고 그렇지 않은 경우, 에너지가 0보다 클 때만 에너지를 1 감소시킨다.
static int nMonster;	Monster클래스의 public에 static을 사용하여 몬스터수를 나타내는 int형 정적 멤버 변수를 선언하였다.
Monster(string n = "푸쉬업몬스터", string i = "♣", int px = 0, int py = 0) : name(n), icon(i), x(px), y(py), nItem(0), nEnergy(INIT_ENERGY) { nMonster++; } ~Monster() { cout << "♣" << name << icon << " 이만 퇴근하겠습니다~~~\n"; nMonster--; }	Monster 생성자 부분의 에너지 초기화는 다음과 같다. Monster 생성자와 소멸자 부분에 멤버 변수 nMonster의 값을 다음과 같이 설정하여 갱신되도록 하였다.
int getEnergy() { return nEnergy; }	Monster클래스의 public에 에너지를 반환하는 함수를 추가하였다.
void print() { cout << "♣" << name << icon << ":" << nItem << " E:" << nEnergy << endl; }	에너지 출력은 다음과 같다.
static void printCount() { cout << "남은 헬스 몬스터 수:" << nMonster << endl; }	정적멤버함수 printCount()를 통해 남은 몬스터의 수를 출력한다.
MonsterWorld.h	
void checkStarvation() { for (int i = 0; i < nMon; i++) { if (pMon[i]->getEnergy() == 0) {	checkStarvation()함수를 통해 몬스터의 에너지가 0이 될 때 해당 배열을 동적 해제하고 마지막 배열

<pre> delete pMon[i]; pMon[i] = pMon[nMon - 1]; nMon--; cout << "Monster 하나가 지방 분해 에너 지를 모두 소비하고 퇴근했습니다." << endl; getchar(); } } </pre>	<p>항목을 해당 배열에 넣고 몬스터 수를 1 감소시키는 코드이다. 이때 화면에 몬스터 하나가 감소되었다는 것을 알리는 문자를 출력하도록 하며, 이를 잠시 멈추어 확인할 수 있도록 getchar()을 추가한다.</p>
<pre> void play(int maxwalk, int wait) { print(); Monster::printCount(); cerr << " 엔터를 누르세요..."; getchar(); for (int i = 0; i < maxwalk; i++) { for (int k = 0; k < nMon; k++) pMon[k]->move(world.Data(), xMax, yMax); nMove++; print(); Monster :: printCount(); checkStarvation(); if (isDone()) break; Sleep(wait); } } </pre>	<p>Monster클래스의 printCount()함수를 호출하여 화면에 남은 몬스터의 수가 출력되도록 하였다.</p> <p>MonsterWorld클래스의 checkStarvation()함수를 호출하여 몬스터의 에너지가 0일 때 동적 해제할 수 있도록 한다.</p>
MonsterWorldGame.cpp	
<pre> int Monster::nMonster = 0; </pre>	<p>다음은 Monster클래스의 정적멤버 변수 선언에 대하여 cpp파일에서의 nMonster 초기화를 나타낸 코드이다.</p>

(3) 다양한 입력에 대한 테스트 결과

//처음 시작 화면, 플랭크 몬스터 퇴근



(4) 코드에 대한 설명 및 해당 문제에 대한 고찰

(5) 이번 과제에 대한 느낀점

이번 과제는 응용 내용이 생각보다 어려워서 이해하기가 힘들어 처음에는 조금 막막하기도 했다. 그러나 막상 문제를 해결하고 나서 보니 생각보다 어렵지 않다고 느꼈다. 그러나 아직 전체적으로 한 번에 정리가 되지 않는 느낌이라 좀 더 노력해야겠다는 생각이 들었다. 앞으로 하나씩 학습하면서 코딩 실력을 향상시켜나가고 싶다.

(6) 궁금한 점이나 건의사항