

2021-1 C++ 프로그래밍 실습과제 10

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

(1) 각 문제에 대한 분석과 및 해결 방법

1. 10.9절의 Monster World 5 프로그램을 두 사람이 경쟁하면서 게임을 할 수 있도록 다음과 같이 확장하라.

- (1) 한 사람(오른쪽)은 화살표 키를 사용하고, 다른 사람(왼쪽)은 상하좌우를 위해 'w', 's', 'a', 'd' 키를 사용한다.
- (2) 왼쪽 사람과 오른쪽 사람의 아이콘은 각각 "좌"와 "우"로 한다.
- (3) Human을 상속한 Tuman 클래스를 만들고, 인간의 움직임만을 위한 move() 함수를 추가한다. 이 함수는 Human::move()과 비슷하지만 입력되는 문자에 따라 움직이는 함수이며, 함수 안에서 kbhit()를 사용하지 않아야 한다.
- (4) 몬스터 월드에 일반 몬스터를 모두 추가한 후 마지막으로 두 개의 Tuman 객체를 추가한다. 이제, pMon[nMon-2]는 왼쪽 경기자가 되고, pMon[nMon-1]은 오른쪽 경기자가 된다.
- (5) MonsterWorld의 play() 함수를 수정하여 두 명 플레이가 되도록 수정한다.

[문제분석 및 해결방법]

교재에 있는 Human 클래스의 자식클래스인 Tuman 클래스를 생성한다. 이때, Human 클래스는 오른쪽 경기자이므로 화살표를 이용하여 움직이고, 자식클래스인 Tuman은 왼쪽 경기자로 하여, WASD 키를 사용하여 움직이도록 한다.

Tuman 클래스에 인간의 움직임만을 위한 move() 함수를 추가한다. 만약 'a'와 'A' 키를 누를 경우 왼쪽, 'd'나 'D' 키를 누를 경우 오른쪽, 'w'나 'W' 키를 누를 경우 위쪽, 's'나 'S' 키를 누를 경우 아래쪽으로 이동할 수 있도록 if문을 사용하여 움직임을 구현하였다. 이때 move() 함수에는 kbhit()는 사용하지 않는다.

MonsterWorldGame.cpp 파일에 main() 함수에서 일반 몬스터 밑에 헬스트레이너 "좌", 헬스트레이너 "우" 아이콘을 가진 Tuman 객체를 add() 함수를 이용해 추가하였다.

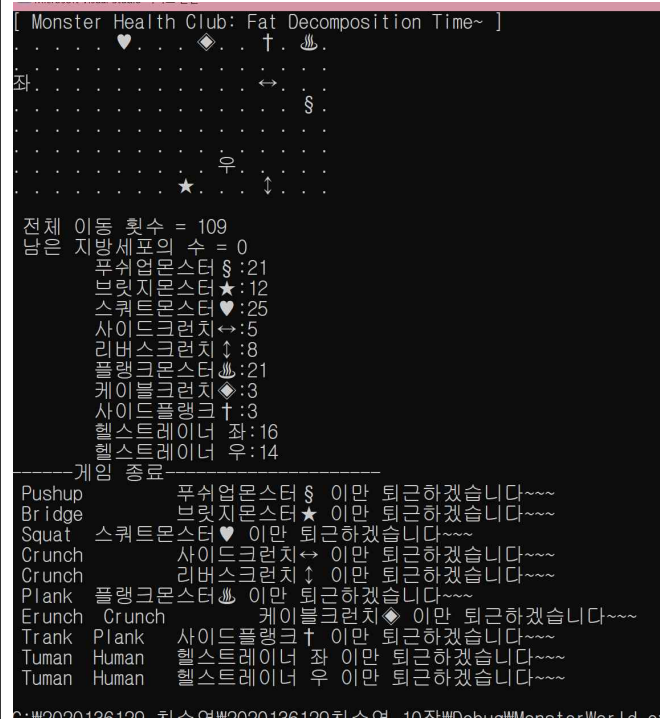

MonsterWorld 헤더파일에 play() 함수에서 이중 for문과 이중 if문을 사용하여 두 명의 헬스트레이너들이 추가되도록 수정한다. 기존 몬스터들은 총 몬스터 수에 -2만큼을 for문으로 반복하여 move를 지정하고, 나머지 두 명의 헬스트레이너들은 키보드가 입력되었을 때, 그 문자의 값이 화살표이면 pMon[nMon-1]로, WASD 키이면 pMon[nMon-2]로 하여 오른쪽 경기자인지 왼쪽 경기자인지를 구분한다.

(2) 자신이 구현한 주요 코드

Human.h	
<pre>class Tuman : public Human { public: Tuman(string n = "헬스트레이너 ", string i = "좌", int px = 0, int py = 0) : Human(n, i, px, py) {} ~Tuman() { cout << " Tuman "; } void move(int** map, int maxx, int maxy, char ch) { if (ch == 'A' ch == 'a' ch == Left) x--; else if (ch == 'D' ch == 'd' ch == Right) x++; else if (ch == 'W' ch == 'w' ch == Up) y--;</pre>	다음은 Tuman 클래스 코드이다. 코드 설명은 위 문제 (1)번의 내용과 같다.

<pre> else if (ch == 'S' ch == 's' ch == Down) y++; else return; clip(maxx, maxy); eat(map); } }; </pre>	
<p>MonsterWorld.h</p> <pre> for (int i = 0; i < maxwalk; i++) { for (int k = 0; k < nMon - 2; k++) pMon[k]->move(world.Data(), xMax, yMax); if (_kbhit()) { unsigned char ch = _getche(); if (ch == 224) { ch = _getche(); ((Tuman*)(pMon[nMon - 1]))->move(world.Data(), xMax, yMax, ch); } else ((Tuman*)(pMon[nMon - 2]))->move(world.Data(), xMax, yMax, ch); } } </pre>	<p>일반 몬스터들과 오른쪽 경기자(화살표), 왼쪽 경기자(WASD)의 move를 다음과 같이 지정하였다.</p>
<p>MonsterWorldGame.cpp</p> <pre> game.add(new Tuman("헬스트레이너 ", "좌", rand() % w, rand() % h)); game.add(new Human("헬스트레이너 ", "우", rand() % w, rand() % h)); </pre>	<p>각각 왼쪽 경기자, 오른쪽 경기자를 cpp파일에 추가하였다.</p>

(3) 다양한 입력에 대한 테스트 결과

 <p>[Monster Health Club: Fat Decomposition Time~]</p> <p>전체 이동 횟수 = 109 남은 지방세포의 수 = 0</p> <ul style="list-style-type: none"> 푸쉬업몬스터 \$:21 브릿지몬스터 ★ :12 스쿼트몬스터 ♥ :25 사이드크런치 ↔ :5 리버스크런치 ↑ :8 플랭크몬스터 ☼ :21 케이블크런치 ◆ :3 사이드플랭크 † :3 헬스트레이너 좌 :16 헬스트레이너 우 :14 <p>게임 종료</p> <p>Pushup 푸쉬업몬스터 \$ 이만 퇴근하겠습니다~~~ Bridge 브릿지몬스터 ★ 이만 퇴근하겠습니다~~~ Squat 스쿼트몬스터 ♥ 이만 퇴근하겠습니다~~~ Crunch 사이드크런치 ↔ 이만 퇴근하겠습니다~~~ Crunch 리버스크런치 ↑ 이만 퇴근하겠습니다~~~ Plank 플랭크몬스터 ☼ 이만 퇴근하겠습니다~~~ Erunch Crunch 케이블크런치 ◆ 이만 퇴근하겠습니다~~~ Trank Plank 사이드플랭크 † 이만 퇴근하겠습니다~~~ Tuman Human 헬스트레이너 좌 이만 퇴근하겠습니다~~~ Tuman Human 헬스트레이너 우 이만 퇴근하겠습니다~~~</p>	 <p>[Monster Health Club: Fat Decomposition Time~]</p> <p>전체 이동 횟수 = 108 남은 지방세포의 수 = 0</p> <ul style="list-style-type: none"> 푸쉬업몬스터 \$:23 브릿지몬스터 ★ :9 스쿼트몬스터 ♥ :33 사이드크런치 ↔ :8 리버스크런치 ↑ :5 플랭크몬스터 ☼ :13 케이블크런치 ◆ :8 사이드플랭크 † :14 헬스트레이너 좌 :9 헬스트레이너 우 :6 <p>게임 종료</p> <p>Pushup 푸쉬업몬스터 \$ 이만 퇴근하겠습니다~~~ Bridge 브릿지몬스터 ★ 이만 퇴근하겠습니다~~~ Squat 스쿼트몬스터 ♥ 이만 퇴근하겠습니다~~~ Crunch 사이드크런치 ↔ 이만 퇴근하겠습니다~~~ Crunch 리버스크런치 ↑ 이만 퇴근하겠습니다~~~ Plank 플랭크몬스터 ☼ 이만 퇴근하겠습니다~~~ Erunch Crunch 케이블크런치 ◆ 이만 퇴근하겠습니다~~~ Trank Plank 사이드플랭크 † 이만 퇴근하겠습니다~~~ Tuman Human 헬스트레이너 좌 이만 퇴근하겠습니다~~~ Tuman Human 헬스트레이너 우 이만 퇴근하겠습니다~~~</p>
---	--

(4) 코드에 대한 설명 및 해당 문제에 대한 고찰

.

(5) 이번 과제에 대한 느낀점

실제로 플레이를 직접 해볼 수 있는 코드가 들어가니 좀 더 게임을 즐기는 맛이 났다. 몬스터와 플레이어 간의 게임 내 직접적인 교류가 있으면 더 재밌을 것 같다.

(6) 궁금한 점이나 건의사항

.