2021-1 C++프로그래밍 실습과제 11

학번	2020136129	이름	최수연

(1) 각 문제에 대한 분석과 및 해결 방법

(2) 자신이 지금까지 구현했던 MonsterWorld에 교재와 같이 Point클래스를 사용하고 동작을 확인한다.

[문제분석 및 해결방법]

교재와 같이 Point 클래스를 사용하여 동작을 확인하였다. Point클래스를 추가하여 좌표 x, y를 Point 객체 하나로 합쳐 사용할 수 있도록 수정하였다. 또한 몬스터마다의 원점으로부터 유클리드 거리를 반환하기 위해 double로 형 변환 연산자를 사용하였다. 반환한 거리는 Monster 클래스에서 double형 total함수를 통해 Monster 소멸자에서 출력되도록 한다.

인덱스 연산자 중복을 사용하여 인덱스 0과 1을 각각 x와 y로 사용한다. 이를 통해 Point클래스 내 private으로 선언된 멤버 변수 x, y를 Monster와 Canvas 클래스 외의 클래스에서 사용할 수 있도록 한다. Canvas와 Monster 클래스는 Point 클래스에서 친구 클래스로 등록하여 Point 클래스의 모든 멤버에 대한 접근이 가능할 수 있도록 하였다.

Point클래스에서 본래 Monster클래스의 clip함수에서 좌표 x, y가 화면 내에 있도록 설정하는 코드를 함수 호출 연산자 중복 함수로 추가하고, 본 Monster클래스의 clip함수에서는 Point에 있는 함수 호출 연산자 중복 함수를 불러올 수 있도록 코드를 수정한다. 또한 Point 클래스에 다양한 산술 연산자 중복 함수를 추가하여, 타 클래스에서 본 함수들을 사용할 수 있도록 한다.

Monster클래스도 교재와 같이 수정하였다. 먼저 Point 클래스의 연산자 중복 함수들을 사용할 수 있도록 하고, x, y 대신 Point 객체를 사용하도록 수정하였으며, 몬스터마다의 거리 계산 및 쉬는 시간을 위한 데이터 멤버와 관련 함수들도 수정하였다. 자세한 코드 설명은 교재 p.524를 참고한다.

Point클래스가 추가되었으므로 자식 클래스에서도 수정을 해주었다. 위와 마찬가지로 x, y 대신 Point 객체를 사용하도록 수정하였으며, Point 함수에서 정의한 인덱스 연산자 중복 함수를 사용하여 멤버 x, y를 접근하도록 수정하였다. 또한 좀비(푸쉬업)몬스터와 신인류 두명을 제외한 나머지 몬스터들에게 쉬는 시간을 부여해야 하므로 현재 쉬는 시간인지 아닌지를 검사하는 코드와 쉬는 시간이 아닐 때 move 함수의 코드가 실행되도록 수정하였다.

Canvas와 MonsterWorld 클래스에서는 draw 함수 부분과 x, y 접근하는 부분의 코드를 Point클래스에 맞게 수정하였다.

(2) 자신이 구현한 주요 코드

Canvas.h

```
#include "Point.h"

class Canvas {
        ...
public:
        ...
        void draw(Point p, string val) {
                if ( p.x >= 0 && p.y >= 0 && p.x < xMax && p.y < yMax)
                        line[p.y].replace(p.x * 2, 2, val);
```

Point클래스에서 Canvas를 친구 클래스로 등록하였기 때문에 Point클래스의 모든 멤버에 대한 접근이 가능하다. 따라서 Canvas클래스의 draw함수에서 x는 p.x로 y는 p.y로

```
        }
        void clear(string val = ". ") {
                for (int y = 0; y < yMax; y++)
                        for (int x = 0; x < xMax; x++)
                                draw(Point(x, y), val);
        }
        ...
};
```

수정한다.
또한 clear함수에서 draw 매개변수 x, y를 Point 객체 하나로 합쳐 수정하였으므로 해당 부분도 수정한다.

Human.h

```
class Human : public Monster {
public:
...
    void move(int** map, int maxx, int maxy) {
        if (_kbhit()) {
            char ch = getDirkey();
            if (ch == Left) p[0]--;
            else if (ch == Right) p[0]++;
            else if (ch == Up) p[1]--;
            else if (ch == Down) p[1]++;
            else return;
            clip(maxx, maxy);
            eat(map);
        }
    }
};
class Tuman : public Human {
public:
...
    void move(int** map, int maxx, int maxy, char ch) {
        if (ch == 'A' || ch == 'a' || ch == Left) p[0]--;
        else if (ch == 'D' || ch == 'd' || ch == Right) p[0]++;
        else if (ch == 'W' || ch == 'w' || ch == Up) p[1]--;
        else if (ch == 'S' || ch == 's' || ch == Down) p[1]++;
        else return;
        clip(maxx, maxy);
        eat(map);
    }
};
```

헬스트레이너 좌, 우는 모두 인간이므로 스스로 컨트롤할 수 있기 때문에 쉬는 시간을 넣지 않았다.

Human 클래스는 Point 클래스의 친구 클래스가 아니기 때문에 p.x, p.y와 같이 접근할 수 없다. 따라서 Human클래스의 x는 p[0], y는 p[1]로 수정한다.
(Point클래스의 인덱스 연산자 중복)

Monster.h

```
#pragma once
#include "Canvas.h"
#include "Point.h"

class Monster {
protected:
```

교재 523p 프로그램11.12와 동일하게 수정하였다. 본 코드에 대한 설명은 위 문제(1)을 참고한다.

```
	string name, icon;
	int nItem;
	Point q, p;
	int nSleep;
	double dist;
	double total;

	void clip(int maxx, int maxy) { p(maxx, maxy); }
	void eat(int** map) {
		if (map[p.y][p.x] == 1) {
			map[p.y][p.x] = 0;
			nItem++;
		}
		dist += (double)(p - q);
		total += (double)(p - q);
		q = p;
		if (dist > 20) {
			dist = 0;
			nSleep = 10;
		}
	}
	bool isSleep() {
		if (nSleep > 0) {
			nSleep--;
			return true;
		}
		else return false;
	}
public:
	Monster(string n = "푸쉬업몬스터", string i = "♤", int x = 0, int y = 0)
		: name(n), icon(i), nItem(0), p(x,y), q(x,y), nSleep(0), dist(0.0), total(0.0) { }
	virtual ~Monster() { cout << icon << nItem << " 거리:" << total << endl; }
	void draw(Canvas& canvas) { canvas.draw(p, icon); }
	virtual void move(int** map, int maxx, int maxy) {
		if (!isSleep()) {
			int num = rand() % 9 + 1;
			p += Point(num % 3 - 1, num / 3 - 1);
			clip(maxx, maxy);
			eat(map);
		}
	}
	void print() { cout << "₩t" << name << icon << ":" << nItem << ":" << nSleep << endl; }
```

```
};
```

MonsterWorld.h

```
#include "Point.h"
...
class MonsterWorld {
...
        void print() {
                ...
                for (int y = 0; y < yMax; y++)
                        for (int x = 0; x < xMax; x++)
                                if (Map(x, y) > 0) canvas.draw(Point(x, y), "■");
...
};
```

x, y 변수를 Point클래스에서 Point객체 하나로 묶어서 코드를 수정하였으므로, MonsterWorld 클래스에서도 해당 부분을 수정해주었다.

VariousMonsters.h

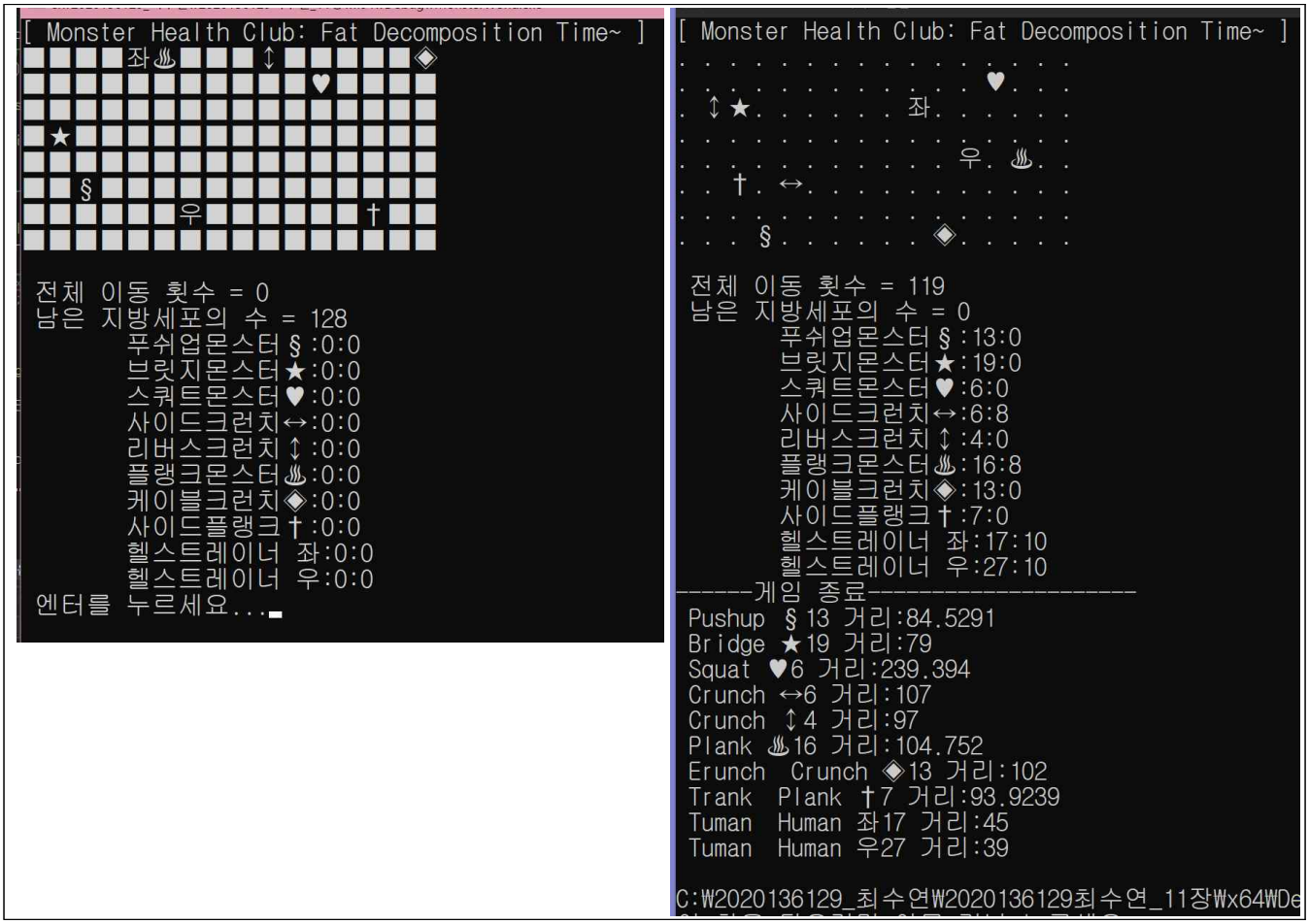
```
class Bridge : public Monster { // 상하좌우로만 움직일 수 있는 몬스터(뱀파이어)
public:
...
    void move(int** map, int maxx, int maxy) {
        if (!isSleep()) {
            switch (rand() % 4) {
            case 0: p[0]++; break;
            case 1: p[0]--; break;
            case 2: p[1]++; break;
            case 3: p[1]--; break;
            }
            clip(maxx, maxy);
            eat(map);
        }
    }
};

class Squat : public Monster { // 랜덤으로 모든 공간으로 이동할 수 있는 몬스터(처녀귀신)
public:
...
    void move(int** map, int maxx, int maxy) {
        if (!isSleep()) {
            p = Point(rand() % maxx, rand() % maxy);
            clip(maxx, maxy);
            eat(map);
        }
    }
};

class Crunch : public Monster { // 상하 또는 좌우로만 움직일 수 있는 몬스터(대
```

Pushup(교재에서는 Zombie)클래스를 제외한 모든 자식 클래스는 먼저 쉬는 시간을 검사하고 쉬는 시간이 아닌 경우에만 기존의 무작위 이동을 진행하였다.

```
륙강시)
protected: bool bHori;
public:
...
    void move(int** map, int maxx, int maxy) {
        if (!isSleep()) {
            int dir = rand() % 2;
            int jump = rand() % 2 + 1;
            if (bHori) p[0] += ((dir == 0) ? -jump : jump);
            else p[1] += ((dir == 0) ? -jump : jump);
            clip(maxx, maxy);
            eat(map);
        }
    }
};

class Plank : public Monster { // 대각선으로만 움직일 수 있는 몬스터(10장 과제 1번, 스몸비)
public:
...
    void move(int** map, int maxx, int maxy) {
        if (!isSleep()) {
            switch (rand() % 4) {
            case 0: p[0]++; p[1]--; break;
            case 1: p[0]++; p[1]++; break;
            case 2: p[0]--; p[1]++; break;
            case 3: p[0]--; p[1]--; break;
            }
            clip(maxx, maxy);
            eat(map);
        }
    }
};

class Erunch : public Crunch { // 일정시간이 지나면 상하 또는 좌우가 바뀌는 몬스터(10장 과제 2번, 샹시)
public:
...
    void move(int** map, int maxx, int maxy) {
        if (!isSleep()) {
            static int num = 0;
            int dir = rand() % 2;
            int jump = rand() % 2 + 1;
            switch (num / 50) {
            case 0: case 2: case 4: case 6: case 8: bHori = true; break;
```

```
                case 1: case 3: case 5: case 7: case 9: bHori = false; break;
                }
                if (bHori) p[0] += ((dir == 0) ? -jump : jump);
                else p[1] += ((dir == 0) ? -jump : jump);
                clip(maxx, maxy);
                eat(map);
                num++;
            }
        }
};

class Trank : public Plank { // 일정시간이 지나면 왼쪽 또는 오른쪽 대각선으로 바꾸는 몬스터(10장 과제 3번)
public:
...
    void move(int** map, int maxx, int maxy) {
        if (!isSleep()) {
            static int num = 0;
            switch (num / 50) {
            case 0: case 2: case 4: case 6: case 8: {
                switch (rand() % 2) {
                case 0: p[0]++; p[1]++; break;
                case 1: p[0]--; p[1]--; break;
                }
            } break;
            case 1: case 3: case 5: case 7: case 9: {
                switch (rand() % 2) {
                case 0: p[0]++; p[1]--; break;
                case 1: p[0]--; p[1]++; break;
                }
            } break;
            }
            clip(maxx, maxy);
            eat(map);
            num++;
        }
    }
};
```

(3) 다양한 입력에 대한 테스트 결과



(4) 코드에 대한 설명 및 해당 문제에 대한 고찰

.

(5) 이번 과제에 대한 느낀점

실제로 학교에 가서 직접 대면수업을 들어보니 온라인으로 2배속으로 항상 수업 들었을 때보다 효율이 좀 떨어져서 과제 하는 과정에서도 살짝 힘들었던 것 같다. 그래도 교수님을 실제로 뵙고 실습하니까 꽤 재밌었다. 이번 11장 내용이 좀 어려워서 실습 때 따라가기 힘들었지만 조교님께서 질문에 친절하게 답변해주셔서 좋았다. 비록 실습 때 제대로 코드를 완성시키지 못했지만 만약에 그 때 통과되었더라면 보고서를 쓰지 않았을 것인데, 막상 보고서를 쓰니까 내용 정리가 잘 돼서 나에게는 당일에 급하게 끝내서 확인받는 것보다 더 효율적이라고 느껴졌다. 교수님께서 왜 보고서 과제를 내주시는지 알 것 같다. 남은 2주도 열심히 C++ 수업을 듣고 시험도 잘 마치고 싶다.

(6) 궁금한 점이나 건의사항

.