

Linux day2 hw3 결과 보고서

2025407012/로봇학부/송연우

목차

1. hpp파일
2. cpp파일
3. 실행 결과

1. hpp 파일

다음은 작성한 헤더파일 코드입니다.

```
1  #include <iostream>
2
3  using namespace std;
4
5  namespace SYU{
6  class Monster;
7
8  class Player
9  {
10
11      public:
12          int x, y, HP, MP;
13          Player();
14          Player(int x, int y);
15          void Attack(Monster &target);
16          void Show_Status();
17          void X_move(int move);
18          void Y_move(int move);
19  };
20  class Monster
21  {
22
23      public:
24          int HP, x, y;
25          Monster();
26          Monster(int x, int y, int HP);
27          int Be_Attacked();
28  };
29  int strlenn(char *str);
30  bool CinSecure(double p);
31  double p;
32  }
33
```

namespace 안에 클래스 Monster와 Player를 선언합니다. Player, Monster는 과제에서 주어진 것과 같이 작성했고, 클래스 밖에서 추가로 strlenn(char *str), bool CinSecure(double p), double p를 선언했습니다. strlenn은 문자열 길이를 반환하는 정수형 함수이고, CinSecure은 정수형 입력을 받는 부분에서 실수 입력을 걸러내기 위한 함수입니다.

2. cpp 파일

다음은 작성한 소스코드입니다.

```
1  #include "hw3.hpp"
2
3
4  namespace SYU{
5
6      Monster::Monster(): x(5), y(4), HP(50) {};
7      Monster::Monster(int x, int y, int HP): x(x), y(y), HP(HP) {};
8  ✓  int Monster::Be_Attacked(){
9      if(this->HP<=0){
10         cout<<"Monster Die!!"<<endl;
11         return 0;
12     }
13     return this->HP;
14
15 }
16
17 Player::Player(): x(0), y(0), HP(100), MP(10) {};
18 Player::Player(int x, int y): x(x), y(y) {};
19 ✓  void Player::Attack(Monster &target){
20     if(target.x==this->x&&target.y==this->y&&MP>0){
21         target.HP-=10;
22         cout<<"공격 성공!"<<endl;
23         cout<<"남은 체력:"<<target.HP<<endl;
24     }
25     else if(this->MP<=1){
26         cout<<"MP부족! 게임 종료..."<<endl;
27     }
28     else{
29         cout<<"공격 실패!"<<endl;
30     }
31     MP--;
32
33 }
```

Monster 클래스의 Monster() 생성자에서는 기본값을 지정합니다. Be_Attacked()함수에서는 Monster클래스에서 생성한 객체의 HP가 0 이하일 때 메시지를 출력하고, 0을 반환합니다. 아니라면 HP값을 반환합니다.

Player 클래스의 Player() 생성자에서는 기본값을 지정합니다. Attack()함수에서는 Monster 클래스의 객체를 주소값으로 받아와 player와 monster의 좌표가 모두 같을 때 monster의 HP값을 10 감소시킵니다. 만약 player의 MP가 1 남은 상태였다면 게임 종료 메시지를 띄웁니다. 위 두 가지 경우가 아니라면 MP가 충분하고 두 객체의 위치가 일치하지 않으므로 공격 실패 메시지를 띄웁니다. 그리고 공격을 1회 실행했으므로 player의 MP값을 1 감소시킵니다.

```
34  void Player::Show_Status(){
35      cout<<"HP:"<<this->HP<<endl;
36      cout<<"MP:"<<this->MP<<endl;
37      cout<<"Position:"<<this->x<<','<<this->y<<endl;
38  }
39  void Player::X_move(int move){
40      this->x+=move;
41      cout<<"X Position "<<move<<" moved!"<<endl;
42  }
43  void Player::Y_move(int move){
44      this->y+=move;
45      cout<<"Y Position "<<move<<" moved!"<<endl;
46  }
```

Show_Status()함수와 X, Y_move()함수입니다. Show_Status()함수에서는 사용자의 HP, MP, 위치를 좌표로 나타내어 출력하며, move함수에서는 move값만큼 사용자의 x, y좌표를 변화시키고 메시지를 출력합니다.

```
48  int strlenn(char *str){
49      int len=0;
50      while(*str){
51          len++;
52          str++;
53      }
54      return len;
55  }
56
57  bool CinSecure(double p){
58      int check=p;
59      if(check!=p){
60          return false;
61      }
62      return true;
63  }
64
65  }
66
```

strlen() 함수와 Cstrcmp()함수입니다. strlen()에서는 문자열의 시작 주소를 받고 주소를 1씩 올리며 NULL문자를 만날 때까지 len을 카운트하고, 반환합니다. Cstrcmp()함수에서는 실수형 매개변수를 받아와 정수형인 check로 변환하고, 정수형과 실수형을 비교해 둘이 다른 값이면 false를, 같으면 true를 반환합니다.

```
67  ✓ int main(){
68      using namespace SYU;
69
70      Player player(0, 0);
71      Monster monster(5, 4, 50);
72
73      char command[100];
74      while(1){
75          cout<<"사용자 스탯 설정-HP: ";
76          cin>>p;
77          if(!CinSecure(p)){
78              cout<<"정수만을 입력할 수 있습니다. 다시 입력하세요"<<endl;
79              continue;
80          }
81          player.HP=p;
82          if(p<=0){
83              cout<<"다시 입력해 주세요."<<endl;
84              continue;
85          }
86          cout<<"사용자 스탯 설정-MP: ";
87          cin>>p;
88          if(!CinSecure(p)){
89              cout<<"정수만을 입력할 수 있습니다. 다시 입력하세요"<<endl;
90              continue;
91          }
92          player.MP=p;
93          if(p<=0){
94              cout<<"다시 입력해 주세요."<<endl;
95              continue;
96          }
97          break;
98      }
99  }
```

main함수에서 처음 설정한 x,y좌표 외 HP값과 MP값을 입력받는 부분입니다. Player, Monster 클래스 객체를 각각 생성한 다음 CinSecure()함수로 정수를 입력받고, HP, MP값을 설정합니다. 정수형 데이터를 저장하고 HP나 MP값이 0이하이면 다시 입력받습니다.

command 문자열은 뒤에서 사용자로부터 명령을 입력받기 위해서 사용합니다.

```

00     while(1){
01         cout<<"Type Command(A/U/D/R/L/S)"<<endl;
02         cin>>command;
03         if(strlen(command)>1){cout<<"Please Enter One Command at Once.";continue;}
04         if(command[0]=='A'){player.Attack(monster);monster.Be_Attacked();}
05         else if(command[0]=='U')player.Y_move(1);
06         else if(command[0]=='D')player.Y_move(-1);
07         else if(command[0]=='R')player.X_move(1);
08         else if(command[0]=='L')player.X_move(-1);
09         else if(command[0]=='S')player.Show_Status();
10         else{
11             cout<<"Wrong Access";continue;
12         }
13         if(monster.HP<=0||player.MP<=0)break;
14
15     }
16
17     return 0;
18 }

```

스텝 설정 후 본격적으로 스크립트가 시작되는 부분입니다. command문자열로 문자를 입력받은 후 strlen()함수로 문자열 길이를 구해 1 이상이면 중복 입력으로 판단하고 오류 메시지를 출력합니다.

사용자가 'A'를 입력하면 Attack() 함수와 Be_Attacked()함수를 실행합니다. 사용자가 'U', 'D', 'R', 'L'를 입력하면 X_move(), Y_move() 함수의 매개변수에 각각 1, -1, 1, -1을 대입해 실행하도록 합니다. 사용자가 'S'를 입력하면 Show_Status()함수를 실행합니다. 다른 값이 입력되었으면 오류 메시지를 출력한 뒤 다시 입력받습니다.

끝으로 monster의 HP가 0이하이거나 player의 MP가 0이하이면 무한반복문을 탈출해 프로그램을 종료합니다.

3. 실행 결과

먼저 정상 작동 시의 실행 결과입니다. 사용자 스탯을 입력받고, 각 커맨드를 입력했을 때 해당 함수가 작동합니다.

```
yu@yu:~/intern_ws/hw3/build$ ./test
사용자 스탯 설정-HP: 100
사용자 스탯 설정-MP: 10
Type Command(A/U/D/R/L/S)
A
공격 실패!
Type Command(A/U/D/R/L/S)
S
HP:100
MP:9
Position:0,0
Type Command(A/U/D/R/L/S)
U
Y Position 1 moved!
Type Command(A/U/D/R/L/S)
D
Y Position -1 moved!
Type Command(A/U/D/R/L/S)
S
HP:100
MP:9
Position:0,0
Type Command(A/U/D/R/L/S)
R
X Position 1 moved!
Type Command(A/U/D/R/L/S)
L
X Position -1 moved!
Type Command(A/U/D/R/L/S)
S
HP:100
MP:9
Position:0,0
```

다음으로, 몬스터 위치인 (5,4)로 이동해 공격 후 프로그램을 종료하겠습니다.

```
Type Command(A/U/D/R/L/S)
U
Y Position 1 moved!
Type Command(A/U/D/R/L/S)
U
Y Position 1 moved!
Type Command(A/U/D/R/L/S)
U
Y Position 1 moved!
Type Command(A/U/D/R/L/S)
U
Y Position 1 moved!
```

```
Type Command(A/U/D/R/L/S)
S
HP:100
MP:9
Position:0,4
Type Command(A/U/D/R/L/S)
R
X Position 1 moved!
Type Command(A/U/D/R/L/S)
R
X Position 1 moved!
Type Command(A/U/D/R/L/S)
R
X Position 1 moved!
Type Command(A/U/D/R/L/S)
R
X Position 1 moved!
Type Command(A/U/D/R/L/S)
R
X Position 1 moved!
Type Command(A/U/D/R/L/S)
S
HP:100
MP:9
Position:5,4
```

몬스터 위치로 이동했습니다.

```

Type Command(A/U/D/R/L/S)
A
공격 성공!
남은 체력:40
Type Command(A/U/D/R/L/S)
A
공격 성공!
남은 체력:30
Type Command(A/U/D/R/L/S)
A
공격 성공!
남은 체력:20
Type Command(A/U/D/R/L/S)
A
공격 성공!
남은 체력:10
Type Command(A/U/D/R/L/S)
A
공격 성공!
남은 체력:0
Monster Die!!
yu@yu:~/intern_ws/hw3/build$ █

```

몬스터의 체력이 표시되고, 0이 되면 프로그램이 종료됩니다.

다음으로 비정상 입력 시 예외처리와 MP고갈 시 프로그램 종료에 대한 실행 결과입니다.

```

yu@yu:~/intern_ws/hw3/build$ ./test
사용자 스탯 설정 -HP: -1
다시 입력해 주세요.
사용자 스탯 설정 -HP: 4.5
정수만을 입력할 수 있습니다. 다시 입력하세요
사용자 스탯 설정 -HP: 10
사용자 스탯 설정 -MP: -1
다시 입력해 주세요.
사용자 스탯 설정 -HP: 10
사용자 스탯 설정 -MP: 4.4
정수만을 입력할 수 있습니다. 다시 입력하세요
사용자 스탯 설정 -HP: 10
사용자 스탯 설정 -MP: 3
Type Command(A/U/D/R/L/S)

```

음수나 실수를 입력하면 오류 메시지가 출력되며 다시 입력받습니다. 아래는 HP를 10, MP를 3으로 설정하고 실행한 결과입니다.


```
Type Command(A/U/D/R/L/S)
^A
Wrong AccessType Command(A/U/D/R/L/S)
^A
Wrong AccessType Command(A/U/D/R/L/S)
A
공격 실패!
Type Command(A/U/D/R/L/S)
A
공격 실패!
Type Command(A/U/D/R/L/S)
A
MP부족! 게임 종료...
yu@yu:~/intern_ws/hw3/build$
```

커맨드에서도 비정상 입력이 감지되면 오류 메시지를 띄웁니다. 공격 3회 후 MP가 0일 때 프로그램이 종료됩니다.