

MonsterWorldGame81.cpp X MonsterWorld81.h X Monster81.h Canvas81.h Matrix.h Solution Explorer

Week8_Homework1 (Global Scope) main

```
1 #include "MonsterWorld81.h"
2 #include <time.h>
3
4 void main()
5 {
6     srand((unsigned int)time(NULL));
7     int w = 16, h = 8;
8
9     MonsterWorld game(w, h);
10    game.add(new Monster("몬스터", "※", rand() % w, rand() % h));
11    game.add(new Monster("도깨비", "§", rand() % w, rand() % h));
12    game.add(new Monster("별그대", "★", rand() % w, rand() % h));
13    game.add(new Monster("고스트", "♥", rand() % w, rand() % h));
14    game.play(500, 10);
15    printf("-----게임 종료-----\n");
16 }
```

C:\Users\user\source\repos\Week8_Homework1\Week8_Homework1.exe

[Monster World (Dynamic World)]

전체 이동 횟수 = 0
남은 아이템 수 = 128
몬스터※:0
도깨비 §:0
별그대★:0
고스트♥:0
엔터를 누르세요...

MonsterWorldGame81.cpp X MonsterWorld81.h X Monster81.h Canvas81.h Matrix.h Solution Explorer

Week8_Homework1 MonsterWorld

```
1 #pragma once
2 #include "Canvas81.h"
3 #include "Monster81.h"
4 #include "Matrix.h"
5 #include <Windows.h>
6 #define MAXMONS 8
7
8 class MonsterWorld
9 {
10 public:
11     Matrix world;
12     int xMax, yMax, nMon, nMove;
13     Monster* pMon[MAXMONS];
14     Canvas canvas;
15
16     int& Map(int x, int y) { return world.elem(x, y); }
17     bool isDone() { return countItems() == 0; }
18     int countItems() {
19         int nItems = 0;
20         for (int y = 0; y < yMax; y++)
21             for (int x = 0; x < xMax; x++)
22                 if (Map(x, y) > 0) nItems++;
23         return nItems;
24     }
25     void print() {
26         canvas.clear(" ");
27         for (int y = 0; y < yMax; y++)
28             for (int x = 0; x < xMax; x++)
29                 if (Map(x, y) > 0) canvas.draw(x, y, "");
30         for (int i = 0; i < nMon; i++)
31             pMon[i]->draw(canvas);
32         canvas.print("[ Monster World (Dynamic World) ]");
33
34         cerr << " 전체 이동 횟수 = " << nMove << endl;
35         cerr << " 남은 아이템 수 = " << countItems() << endl;
36         for (int i = 0; i < nMon; i++) pMon[i]->print();
37     }
38 };
```

C:\Users\user\source\repos\Week8_Homework1\Week8_Homework1.exe

[Monster World (Dynamic World)]

전체 이동 횟수 = 18
남은 아이템 수 = 76
몬스터※:15
도깨비 §:11
별그대★:13
고스트♥:13

MonsterWorldGame81.cpp X MonsterWorld81.h X Monster81.h Canvas81.h Matrix.h Solution Explorer

Week8_Homework1 MonsterWorld

```
34 cerr << " 남은 아이템 수 = " << countItems() << endl;
35 for (int i = 0; i < nMon; i++) pMon[i]->print();
36
37 public:
38     MonsterWorld(int w, int h) : world(h, w), canvas(w, h), xMax(w), yMax(h) {
39         nMon = 0;
40         nMove = 0;
41         for (int y = 0; y < yMax; y++)
42             for (int x = 0; x < xMax; x++) Map(x, y) = 1;
43     }
44     ~MonsterWorld() {
45         for (int i = 0; i < nMon; i++)
46             delete pMon[i];
47     }
48     void add(Monster* m) { if (nMon < MAXMONS) pMon[nMon++] = m; }
49     void play(int maxwalk, int wait) {
50         print();
51         cerr << " 엔터를 누르세요... ";
52         getchar();
53         for (int i = 0; i < maxwalk; i++)
54         {
55             for (int k = 0; k < nMon; k++)
56                 pMon[k]->move(world.Data(), xMax, yMax);
57             nMove++;
58             print();
59             if (isDone()) break;
60             Sleep(wait);
61         }
62     }
63 }
```

C:\Users\user\source\repos\Week8_Homework1\Week8_Homework1.exe

[Monster World (Dynamic World)]

전체 이동 횟수 = 137
남은 아이템 수 = 21
몬스터※:26
도깨비 §:18
별그대★:19
고스트♥:44

The screenshot displays a C++ development environment with two main panes. The left pane shows the source code for a program titled "MonsterWorldGame81.cpp". The code defines a "Monster" class with attributes like name, icon, x, y, nItem, and methods such as clip, eat, draw, and move. It also includes a main function that initializes a monster and draws it on a canvas.

The right pane shows the output of the program, titled "C:\Users\Wuser\source\repos\Week8_Homework1\Debug\Week8_Homework1.exe". The output displays a grid representing the game world, where different symbols represent various elements like monsters, items, and terrain. Below the grid, statistics are listed:

```
[ Monster World (Dynamic World) ]  
.  
. . . . . ■■■■■■ . ■  
. . . § . . . ■■■■ .  
. . . . . ■■ .  
. . . . . ♥ .  
. . . . .  
■. ※★ . . . . .
```

전체 이동 횟수 = 269
남은 아이템 수 = 15
몬스터※:28
도깨비§:18
별그대★:19
고스트♥:48

```

1  #include <iostream>
2  #include <vector>
3  #include <string>
4  #include <map>
5  #include <algorithm>
6  #include <random>
7  #include <ctime>
8  #include <chrono>
9  #include <thread>
10 #include <mutex>
11 #include <atomic>
12 #include <memory>
13 #include <functional>
14 #include <string_view>
15 #include <variant>
16 #include <optional>
17 #include <any>
18 #include <tuple>
19 #include <utility>
20 #include <array>
21 #include <deque>
22 #include <list>
23 #include <queue>
24 #include <stack>
25 #include <vector>
26 #include <map>
27 #include <set>
28 #include <unordered_map>
29 #include <unordered_set>
30 #include <string>
31 #include <string_view>
32 #include <memory>
33 #include <memory_order>
34 #include <atomic>
35 #include <mutex>
36 #include <thread>
37 #include <chrono>
38 #include <ctime>
39 #include <random>
40 #include <algorithm>
41 #include <functional>
42 #include <string_view>
43 #include <variant>
44 #include <optional>
45 #include <any>
46 #include <tuple>
47 #include <utility>
48 #include <array>
49 #include <deque>
50 #include <list>
51 #include <queue>
52 #include <stack>
53 #include <vector>
54 #include <map>
55 #include <set>
56 #include <unordered_map>
57 #include <unordered_set>
58 #include <string>
59 #include <string_view>
60 #include <memory>
61 #include <memory_order>
62 #include <atomic>
63 #include <mutex>
64 #include <thread>
65 #include <chrono>
66 #include <ctime>
67 #include <random>
68 #include <algorithm>
69 #include <functional>
70 #include <string_view>
71 #include <variant>
72 #include <optional>
73 #include <any>
74 #include <tuple>
75 #include <utility>
76 #include <array>
77 #include <deque>
78 #include <list>
79 #include <queue>
80 #include <stack>
81 #include <vector>
82 #include <map>
83 #include <set>
84 #include <unordered_map>
85 #include <unordered_set>
86 #include <string>
87 #include <string_view>
88 #include <memory>
89 #include <memory_order>
90 #include <atomic>
91 #include <mutex>
92 #include <thread>
93 #include <chrono>
94 #include <ctime>
95 #include <random>
96 #include <algorithm>
97 #include <functional>
98 #include <string_view>
99 #include <variant>
100 #include <optional>
101 #include <any>
102 #include <tuple>
103 #include <utility>
104 #include <array>
105 #include <deque>
106 #include <list>
107 #include <queue>
108 #include <stack>
109 #include <vector>
110 #include <map>
111 #include <set>
112 #include <unordered_map>
113 #include <unordered_set>
114 #include <string>
115 #include <string_view>
116 #include <memory>
117 #include <memory_order>
118 #include <atomic>
119 #include <mutex>
120 #include <thread>
121 #include <chrono>
122 #include <ctime>
123 #include <random>
124 #include <algorithm>
125 #include <functional>
126 #include <string_view>
127 #include <variant>
128 #include <optional>
129 #include <any>
130 #include <tuple>
131 #include <utility>
132 #include <array>
133 #include <deque>
134 #include <list>
135 #include <queue>
136 #include <stack>
137 #include <vector>
138 #include <map>
139 #include <set>
140 #include <unordered_map>
141 #include <unordered_set>
142 #include <string>
143 #include <string_view>
144 #include <memory>
145 #include <memory_order>
146 #include <atomic>
147 #include <mutex>
148 #include <thread>
149 #include <chrono>
150 #include <ctime>
151 #include <random>
152 #include <algorithm>
153 #include <functional>
154 #include <string_view>
155 #include <variant>
156 #include <optional>
157 #include <any>
158 #include <tuple>
159 #include <utility>
160 #include <array>
161 #include <deque>
162 #include <list>
163 #include <queue>
164 #include <stack>
165 #include <vector>
166 #include <map>
167 #include <set>
168 #include <unordered_map>
169 #include <unordered_set>
170 #include <string>
171 #include <string_view>
172 #include <memory>
173 #include <memory_order>
174 #include <atomic>
175 #include <mutex>
176 #include <thread>
177 #include <chrono>
178 #include <ctime>
179 #include <random>
180 #include <algorithm>
181 #include <functional>
182 #include <string_view>
183 #include <variant>
184 #include <optional>
185 #include <any>
186 #include <tuple>
187 #include <utility>
188 #include <array>
189 #include <deque>
190 #include <list>
191 #include <queue>
192 #include <stack>
193 #include <vector>
194 #include <map>
195 #include <set>
196 #include <unordered_map>
197 #include <unordered_set>
198 #include <string>
199 #include <string_view>
200 #include <memory>
201 #include <memory_order>
202 #include <atomic>
203 #include <mutex>
204 #include <thread>
205 #include <chrono>
206 #include <ctime>
207 #include <random>
208 #include <algorithm>
209 #include <functional>
210 #include <string_view>
211 #include <variant>
212 #include <optional>
213 #include <any>
214 #include <tuple>
215 #include <utility>
216 #include <array>
217 #include <deque>
218 #include <list>
219 #include <queue>
220 #include <stack>
221 #include <vector>
222 #include <map>
223 #include <set>
224 #include <unordered_map>
225 #include <unordered_set>
226 #include <string>
227 #include <string_view>
228 #include <memory>
229 #include <memory_order>
230 #include <atomic>
231 #include <mutex>
232 #include <thread>
233 #include <chrono>
234 #include <ctime>
235 #include <random>
236 #include <algorithm>
237 #include <functional>
238 #include <string_view>
239 #include <variant>
240 #include <optional>
241 #include <any>
242 #include <tuple>
243 #include <utility>
244 #include <array>
245 #include <deque>
246 #include <list>
247 #include <queue>
248 #include <stack>
249 #include <vector>
250 #include <map>
251 #include <set>
252 #include <unordered_map>
253 #include <unordered_set>
254 #include <string>
255 #include <string_view>
256 #include <memory>
257 #include <memory_order>
258 #include <atomic>
259 #include <mutex>
260 #include <thread>
261 #include <chrono>
262 #include <ctime>
263 #include <random>
264 #include <algorithm>
265 #include <functional>
266 #include <string_view>
267 #include <variant>
268 #include <optional>
269 #include <any>
270 #include <tuple>
271 #include <utility>
272 #include <array>
273 #include <deque>
274 #include <list>
275 #include <queue>
276 #include <stack>
277 #include <vector>
278 #include <map>
279 #include <set>
280 #include <unordered_map>
281 #include <unordered_set>
282 #include <string>
283 #include <string_view>
284 #include <memory>
285 #include <memory_order>
286 #include <atomic>
287 #include <mutex>
288 #include <thread>
289 #include <chrono>
290 #include <ctime>
291 #include <random>
292 #include <algorithm>
293 #include <functional>
294 #include <string_view>
295 #include <variant>
296 #include <optional>
297 #include <any>
298 #include <tuple>
299 #include <utility>
300 #include <array>
301 #include <deque>
302 #include <list>
303 #include <queue>
304 #include <stack>
305 #include <vector>
306 #include <map>
307 #include <set>
308 #include <unordered_map>
309 #include <unordered_set>
310 #include <string>
311 #include <string_view>
312 #include <memory>
313 #include <memory_order>
314 #include <atomic>
315 #include <mutex>
316 #include <thread>
317 #include <chrono>
318 #include <ctime>
319 #include <random>
320 #include <algorithm>
321 #include <functional>
322 #include <string_view>
323 #include <variant>
324 #include <optional>
325 #include <any>
326 #include <tuple>
327 #include <utility>
328 #include <array>
329 #include <deque>
330 #include <list>
331 #include <queue>
332 #include <stack>
333 #include <vector>
334 #include <map>
335 #include <set>
336 #include <unordered_map>
337 #include <unordered_set>
338 #include <string>
339 #include <string_view>
3
```

The screenshot displays a C++ development environment with two main windows.

Left Window (Code Editor):

- Title Bar:** Week8_Homework1 | Canvas
- Code:**

```
1 #pragma once  
2 #include <iostream>  
3 #include <string>  
4 #define MAXLINES 100  
5 using namespace std;  
6  
7 class Canvas  
8 {  
9     string line[MAXLINES];  
10    int xMax, yMax;  
11 public:  
12    Canvas(int nx = 10, int ny = 10) : xMax(nx), yMax(ny) {  
13        for (int y = 0; y < yMax; y++)  
14            line[y] = string(xMax * 2, ' ');  
15    }  
16    void draw(int x, int y, string val) {  
17        if (x >= 0 && y >= 0 && x < xMax && y < yMax)  
18            line[y].replace(x * 2, 2, val);  
19    }  
20    void clear(string val = " ") {  
21        for (int y = 0; y < yMax; y++)  
22            for (int x = 0; x < xMax; x++)  
23                draw(x, y, val);  
24    }  
25    void print(const char* title = "<My Canvas>") {  
26        system("cls");  
27        cout << title << endl;  
28        for (int y = 0; y < yMax; y++)  
29            cout << line[y] << endl;  
30        cout << endl;  
31    }  
32 };  
33
```

Right Window (Microsoft Visual Studio Debug Console):

- Title Bar:** Microsoft Visual Studio Debug Console
- Content:**

```
[ Monster World (Dynamic World) ]  
★.....※.....♥..  
. . . . .  
.  §  . . . . .  
. . . . .  
전체 이동 횟수 = 419  
남은 아이템 수 = 0  
몬스터※:30  
도깨비§:19  
별그대★:19  
고스트♥:60  
-----게임 종료-----  
몬스터※물러갑니다~~~  
도깨비§물러갑니다~~~  
별그대★물러갑니다~~~  
고스트♥물러갑니다~~~  
  
C:\Users\User\source\repos\Week8_Homework1\wx64\Debug\Week8_Homework1.exe (process 20212) exited wi  
Press any key to close this window . . .
```

Status Bar (Bottom Left): 79 % No issues found



