

```
1  #include <iostream>
2
3  #define M 16
4  #define N 12
5
6
7  void drawField();
8  void drawNewPosition(char);
9
10
11 typedef struct {
12     char f;
13     bool isWall;
14 } field;
15
16
17 ///////////////////////////////////////////////////
18 //                                SETUP
19 ///////////////////////////////////////////////////
20
21 // Initial Position
22 int x = 5, y = 5;
23
24 // Define a playing area
25 field playground[N][M];
26
27
28 ///////////////////////////////////////////////////
29 //                                GAME ON!
30 ///////////////////////////////////////////////////
31 // Note that this is not optimized: there's no control over what is given
32 // as input and what the program should do in case of unexpected commands
33 // though this probably won't cause problems
34
35
36 int main() {
37     // Let's get started
38     drawField();
39
40     // First move
41     char c;
42     std::cin >> c;
43
44     // Check the first move and continue to receive new moves until
45     // the end of the game (command 'q')
46     while (c != 'q') {
47         if (c == 'l' || c == 'r' || c == 'u' || c == 'd')
48             drawNewPosition(c);
49         std::cin >> c;
50     }
51
52     return 0;
53 }
54
55
56
57
58
```

```
59 // DRAW INITIAL STATE OF THE GAME
60 void drawField() {
61     for (int i = 0; i < N; i++) {
62         for (int j = 0; j < M; j++) {
63             // In the borders of the field, the following will result true.
64             // Note that there's an opening at the top side of the field
65             // (at the fourth column).
66             playground[i][j].isWall = ((i == 0 && j != 3) || i == (N - 1) ||
67                                         (j == 0) || j == (M - 1));
68
69             // Drawing
70             if (playground[i][j].isWall) // Wall
71                 playground[i][j].f = '*';
72             else if (i == x && j == y) // Initial Position
73                 playground[i][j].f = 'O';
74             else
75                 playground[i][j].f = ' '; // Inside field
76
77             // Printing
78             std::cout << playground[i][j].f;
79         }
80         std::cout << "\n";
81     }
82 }
83
84
85 // UPDATE STATE OF THE GAME
86 void drawNewPosition(char c) {
87     // Free old position
88     playground[x][y].f = ' ';
89
90     // Check that the player remain into the field
91     // after moving
92     if (c == 'l' && y < M - 1 && y > 1 && x != 0)
93         y--;
94     else if (c == 'r' && y < M - 2 && y > 0 && x != 0)
95         y++;
96     else if (c == 'u' && x < N && (x > 0 && y == 3))
97         x--;
98     else if (c == 'u' && x < N && (x > 1 && y != 3))
99         x--;
100     else if (c == 'd' && x < N - 2 && (x >= 0))
101         x++;
102
103     // Draw new position
104     playground[x][y].f = 'O';
105
106     // Print new state of the game
107     for (int i = 0; i < N; i++) {
108         for (int j = 0; j < M; j++)
109             std::cout << playground[i][j].f;
110         std::cout << "\n";
111     }
112 }
```