



```

1  #include <stdio.h>
2  #include <windows.h>
3  #include <conio.h>
4  #include <time.h>
5
6  int score = 0;
7
8  int max_bullet = 5;
9  int pos_x[5], pos_y[5];
10 int count = 0;
11
12 void setcolor(int fg, int bg)
13 {
14     HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
15     SetConsoleTextAttribute(hConsole, bg * 16 + fg);
16 }
17
18 void gotoxy(int x, int y)
19 {
20     COORD c = {x, y};
21     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
22 }
23
24 void draw_ship(int x, int y)
25 {
26     gotoxy(x, y);
27     setcolor(2, 4);
28     printf("<-0->");
29 }
30
31 void erase_ship(int x, int y)
32 {
33     gotoxy(x, y);
34     setcolor(0, 0);
35     printf(" ");
36 }
37
38 void setcursor(bool visible)
39 {
40     HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
41     CONSOLE_CURSOR_INFO lpCursor;
42     lpCursor.bVisible = visible;
43     lpCursor.dwSize = 20;
44     SetConsoleCursorInfo(console, &lpCursor);
45 }
46
47 void draw_bullet(int x, int y)
48 {
49     gotoxy(x, y);
50     setcolor(2, 0);
51     printf("^");
52 }
53
54 void erase_bullet(int x, int y)
55 {
56     gotoxy(x, y);
57     setcolor(0, 0);
58     printf(" ");
59 }
60
61 void draw_star(int x, int y)
62 {
63     gotoxy(x, y);
64     setcolor(7, 0);
65     printf("***");
66 }
67
68 void erase_star(int x, int y)
69 {
70     gotoxy(x, y);
71     printf(" ");
72 }
73
74 void draw_score(int x, int y)
75 {
76     gotoxy(x, y);
77     setcolor(7, 0);
78     printf("Score : %d", score);
79 }

```

```

80
81 char cursor(int x, int y)
82 {
83     HANDLE hStd = GetStdHandle(STD_OUTPUT_HANDLE);
84     char buf[2];
85     COORD c = {x, y};
86     DWORD num_read;
87     if (!ReadConsoleOutputCharacter(hStd, (LPTSTR)buf, 1, c, (LPDWORD)&num_read))
88     {
89         return '\0';
90     }
91     else
92     {
93         return buf[0];
94     }
95 }
96
97 int main()
98 {
99     srand(time(NULL));
100     int score_x = 82, score_y = 1;
101     int WIDTH = 80, HEIGHT = 20;
102     int direction = 0;
103     setcursor(0);
104     char ch = ' ';
105     int x = 38, y = 20;
106
107     draw_score(score_x, score_y);
108     draw_ship(x, y);
109
110     for (int i = 0; i < 20; i++)
111     {
112         draw_star(10 + rand() % 69, 2 + rand() % 4);
113     }
114
115     do
116     {
117         int bullet_state = 0;
118         if (kbhit())
119         {
120             ch = getch();
121             if (ch == 'a')
122                 direction = -1;
123
124             if (ch == 'd')
125                 direction = 1;
126
127             if (ch == 's')
128                 direction = 0;
129
130             if (ch == ' ')
131                 bullet_state = 1;
132
133             fflush(stdin);
134         }
135         if (direction != 0)
136         {
137             if (x + direction > WIDTH - 5)
138                 direction = 0;
139             if (x + direction < 0)
140                 direction = 0;
141
142             erase_ship(x, y);
143             x += direction;
144             draw_ship(x, y);
145         }
146         if (bullet_state && pos_y[count] < 0)
147         {
148             if (y > 0)
149             {
150                 pos_x[count] = x + 2;
151                 pos_y[count] = y - 1;
152                 count++;
153                 count %= max_bullet;
154                 draw_bullet(x + 2, y - 1);
155                 Beep(700, 25);
156
157                 if (cursor(pos_x[count], pos_y[count] - 1) == '*')
158                 {
159                     draw_star(10 + rand() % 69, 2 + rand() % 4);
160                     score += 1;
161                     draw_score(score_x, score_y);
162                     Beep(700, 25);
163                 }
164             }
165         }
166
167         for (int i = 0; i < max_bullet; i++)
168         {
169             if (pos_y[i] < 0)
170                 continue;
171             erase_bullet(pos_x[i], pos_y[i]);
172             pos_y[i]--;
173             if (pos_y[i] >= 0)
174                 draw_bullet(pos_x[i], pos_y[i]);
175             if (cursor(pos_x[i], pos_y[i] - 1) == '*')
176             {
177                 draw_star(10 + rand() % 69, 2 + rand() % 4);
178                 score += 1;
179                 draw_score(score_x, score_y);
180                 Beep(700, 25);
181             }
182         }
183
184         Sleep(100);
185     } while (ch != 'x');
186     return 0;
187 }

```