

```

1  #include<stdio.h>
2  #include<windows.h>
3  #include<conio.h>
4  #include<time.h>
5
6  void setcolor(int fg, int bg)
7  {
8      HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
9      SetConsoleTextAttribute(hConsole, bg * 16 + fg);
10 }
11 void gotoxy(int x, int y)
12 {
13     COORD c = { x, y };
14     SetConsoleCursorPosition(
15         GetStdHandle(STD_OUTPUT_HANDLE), c);
16 }
17 void setcursor(bool visible)
18 {
19     HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
20     CONSOLE_CURSOR_INFO lpCursor;
21     lpCursor.bVisible = visible;
22     lpCursor.dwSize = 20;
23     SetConsoleCursorInfo(console, &lpCursor);
24 }
25 void draw_ship(int x, int y)
26 {
27     gotoxy(x, y);
28     setcolor(2, 4);
29     printf(" <-0-> ");
30 }
31 void erase_ship(int x, int y)
32 {
33     gotoxy(x, y);
34     setcolor(0, 0);
35     printf("      ");
36     setcolor(2, 4);
37 }
38 void draw_bullet(int x, int y)
39 {
40     gotoxy(x, y);
41     printf("| ");
42 }
43 void erase_bullet(int x, int y)
44 {
45     gotoxy(x, y);

```

```

46         setcolor(0, 0);
47         printf(" ");
48         setcolor(3, 5);
49     }
50 void draw_star(int x, int y)
51 {
52     gotoxy(x, y);
53     setcolor(2, 0);
54     printf("*");
55 }
56 void show_score(int x, int y, int s)
57 {
58     gotoxy(x, y);
59     setcolor(4, 7);
60     printf("%d", s);
61 }
62 char cursor(int x, int y) {
63     HANDLE hStd = GetStdHandle(STD_OUTPUT_HANDLE);
64     char buf[2]; COORD c = { x,y }; DWORD num_read;
65     if (
66         !ReadConsoleOutputCharacter(hStd, (LPTSTR)buf, 1, c, (LPDWORD)&num_read))
67         return '\0';
68     else
69         return buf[0];
70 }
71
72
73 int main()
74 {
75     char ch = 'k', move = 'k';
76     int x = 38, y = 20, count = 0, score = 0;
77     int xb[20][], yb[20][], slot = 0;
78     bool bullet[20][];
79
80     srand(time(NULL));
81     for (int i = 0; i < 20; i++) { draw_star(rand() % 60 + 10, rand() % 4 + 2); }
82     draw_star(42, 6);
83     draw_star(41, 6);
84     draw_star(40, 6);
85
86     setcursor(0);
87     setcolor(2, 4);
88     draw_ship(x, y);
89
90     do {

```

```
91  if (_kbhit()) {
92      ch = _getch();
93      if (ch == 'a') {
94          move = 'a';
95      }
96      if (ch == 'd') {
97          move = 'd';
98      }
99      if (ch == 'w') {
100          move = 'w';
101      }
102      if (ch == 's') {
103          move = 's';
104      }
105      if (ch == 'c') {
106          move = 'c';
107      }
108      if (ch == ' ' && count < 5) {
109          Beep(700, 50);
110          count++;
111          slot = (slot + 1) % 5;
112          bullet[slot] = true;
113          xb[slot] = x + 2;
114          yb[slot] = y;
115      }
116  }
117  if (move == 'a') {
118      if (x - 1 > 0) {
119          erase_ship(x, y);
120          draw_ship(--x, y);
121      }
122  }
123  if (move == 'd') {
124      if (x + 1 < 70) {
125          erase_ship(x, y);
126          draw_ship(++x, y);
127      }
128  }
129  if (move == 'w') {
130      if (y - 1 > 0) {
131          erase_ship(x, y);
132          draw_ship(x, --y);
133      }
134  }
135  if (move == 's') {
```

```

137         erase_ship(x, y);
138         draw_ship(x, ++y);
139     }
140 }
141 for (int i = 0; i < 5; i++){
142     if (bullet[i] == true) {
143         erase_bullet(xb[i], yb[i]);
144         if (yb[i] > 0 && yb[i] < 40) {
145             draw_bullet(xb[i], --yb[i]);
146             draw_ship(x, y);
147             if (cursor(xb[i], yb[i] - 1) == '*' || cursor(xb[i] + 1, yb[i] - 1) == '*' || cursor(xb[i] + 2, yb[i] - 1) == '*') {
148                 if (cursor(xb[i], yb[i] - 1) == '*') {
149                     draw_star(rand() % 60 + 10, rand() % 4 + 2);
150                     score++;
151                 }
152                 if (cursor(xb[i] + 1, yb[i] - 1) == '*') {
153                     draw_star(rand() % 60 + 10, rand() % 4 + 2);
154                     score++;
155                 }
156                 if (cursor(xb[i] + 2, yb[i] - 1) == '*') {
157                     draw_star(rand() % 60 + 10, rand() % 4 + 2);
158                     score++;
159                 }
160                 erase_bullet(xb[i], yb[i]);
161                 erase_bullet(xb[i], yb[i] - 1);
162                 Beep(200, 50);
163                 count--;
164                 bullet[i] = false;
165             }
166         }
167     }
168     if (yb[i] < 1) {
169         erase_bullet(xb[i], yb[i]);
170         count--;
171         bullet[i] = false;
172     }
173 }
174
175 show_score(69, 1, score);
176 fflush(stdin);
177 Sleep(100);
178 } while (ch != 'x');
179 setcolor(7, 0);
180 return 0;
181 }

```