```cpp
#define _CRT_SECURE_NO_WARNINGS 1
#include<stdio.h>
#include<windows.h>
#include<time.h>
#include<conio.h>
#include<thread>

int status[5];
int xb[5], yb[5];

void setcolor(int fg, int bg)
{
    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hConsole, bg * 16 + fg);
}

void gotoxy(int x, int y)
{
    COORD c = { x, y };
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), c);
}

void draw_ship(int x, int y)
{
    gotoxy(x, y);
    setcolor(2, 4);
    printf(" <-0-> ");
}

void erase_ship(int x, int y)
{
    gotoxy(x, y);
    setcolor(0, 0);
    printf("       ");
}

struct bullet
{
    bool status;
    int x, y;
};

struct bullet A[5];

void draw_bullet(int x, int y)
{
    gotoxy(x, y);
    setcolor(7, 0);
    printf("^");
```

```
50  }
51
52  void clear_bullet(int x, int y)
53  {
54      gotoxy(x, y);
55      setcolor(0, 0);
56      printf(" ");
57  }
58
59  void draw_star(int x, int y)
60  {
61      gotoxy(x, y);
62      setcolor(7, 0);
63      printf("*");
64  }
65
66  void clear_star(int x, int y)
67  {
68      gotoxy(x,y);
69      setcolor(0, 0);
70      printf(" ");
71  }
72
73  void score(int z, int x, int y)
74  {
75      gotoxy(x, y);
76      printf("%d", z);
77  }
78
79  void setcursor(bool visible)
80  {
81      HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
82      CONSOLE_CURSOR_INFO lpCursor;
83      lpCursor.bVisible = visible;
84      lpCursor.dwSize = 20;
85      SetConsoleCursorInfo(console, &lpCursor);
86  }
87
88  char cursor(int x, int y)
89  {
90      HANDLE hStd = GetStdHandle(STD_OUTPUT_HANDLE);
91      char buf[2];
92      COORD c = { x,y };
93      DWORD num_read;
94      if (!ReadConsoleOutputCharacter(hStd, (LPTSTR)buf, 1, c, (LPDWORD) &
          num_read))
95          return '\0';
96      else
97          return buf[0];
```

```
 98  }
 99
100  int main()
101  {
102      int sc = 0, hit = 0;
103      int sx, sy;
104      char ch = ' ';
105      char direction = ' ';
106      int x = 38, y = 20;
107      int bx, by, i;
108      int bullet = 0;
109      setcursor(0);
110      draw_ship(x, y);
111
112      srand(time(NULL));
113      for (int i = 1; i <= 20; i++)
114      {
115          sy = 10 + rand() % 61; /*[10-70] minimun = 10 range = 61*/
116          sx = 2 + rand() % 4;   /*[2-5] minimun = 2 range = 4*/
117          draw_star(sx, sy);
118      }
119
120      do
121      {
122          if (_kbhit())
123          {
124              ch = _getch();
125              if (ch == 'a')
126              {
127                  direction = 'a';
128              }
129              if (ch == 'd')
130              {
131                  direction = 'd';
132              }
133              if (ch == 's')
134              {
135                  direction = 's';
136              }
137              if (ch == ' ')
138              {
139                  for (int i = 0; i < 5; i++)
140                  {
141                      if (A[i].status == false)
142                      {
143                          A[i].status = true;
144                          A[i].x = x + 3;
145                          A[i].y = y - 1;
146                          draw_bullet(A[i].x, A[i].y);
```

```
147                         std::thread q(Beep, 360, 1900);
148                         q.detach();
149                         break;
150                     }
151                 }
152             }
153             fflush(stdin);
154         }
155         if (direction == 'a' && x > 0)
156         {
157             erase_ship(x, y);
158             draw_ship(--x, y);
159         }
160         if (direction == 'd' && x < 73)
161         {
162             erase_ship(x, y);
163             draw_ship(++x, y);
164         }
165         for (int i = 0; i < 5; i++)
166         {
167             if (A[i].status == true)
168             {
169                 if (A[i].y == 2)
170                 {
171                     A[i].status = false;
172                     clear_bullet(A[i].x, A[i].y);
173                 }
174 
175                 else
176                 {
177                     if (cursor(A[i].x, A[i].y - 1) == '*')
178                     {
179                         hit = 1;
180                     }
181                     clear_bullet(A[i].x, A[i].y);
182                     draw_bullet(A[i].x, --A[i].y);
183                 }
184                 if (hit == 1)
185                 {
186                     sc += 1;
187                     clear_bullet(A[i].x, A[i].y);
188                     sx = 10 + rand() % 61;
189                     sy = 2 + rand() % 4;
190                     draw_star(sx, sy);
191                     A[i].status = false;
192                     hit = 0;
193                     std::thread q(Beep, 400, 100);
194                     q.detach();
195                 }
```

```
196             }
197         }
198         score(sc, 77, 0);
199         Sleep(100);
200     } while (ch != 'x');
201
202     return 0;
203 }
```