

-Source Code

```
#define CRT_NO_SECURE_WARNINGS 1
#include <stdio.h>
#include <windows.h>
#include <time.h>
#include <conio.h>

#define scount 80
#define screen_x 80
#define screen_y 25

HANDLE wHnd;
HANDLE rHnd;
DWORD fdwMode;
CHAR_INFO consoleBuffer[screen_x * screen_y];
COORD bufferSize = { screen_x, screen_y };
COORD characterPos = { 0, 0 };
SMALL_RECT windowSize = { 0, 0, screen_x - 1, screen_y - 1 };
COORD star[scount];
COORD ship;

int sposX = screen_x / 2;
int sposY = screen_y - 1;
int cnt = 0;
bool play = true;
int color = 7;

int setConsole(int x, int y)
{
    wHnd = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleWindowInfo(wHnd, TRUE, &windowSize);
    SetConsoleScreenBufferSize(wHnd, bufferSize);
    return 0;
}

int setMode()
{
    rHnd = GetStdHandle(STD_INPUT_HANDLE);
    fdwMode = ENABLE_EXTENDED_FLAGS | ENABLE_WINDOW_INPUT |
        ENABLE_MOUSE_INPUT;
    SetConsoleMode(rHnd, fdwMode);
    return 0;
}

void clear_buffer()
{
    for (int y = 0; y < screen_y; ++y) {
        for (int x = 0; x < screen_x; ++x) {
            consoleBuffer[x + screen_x * y].Char.AsciiChar = ' ';
            consoleBuffer[x + screen_x * y].Attributes = 7;
        }
    }
}

void fill_buffer_to_console()
{
    WriteConsoleOutputA(wHnd, consoleBuffer, bufferSize, characterPos,
```

```

        &windowSize);
}
void init_star()
{
    for (int i = 0; i < scount; i++)
    {
        star[i] = { SHORT(rand() % screen_x), SHORT(rand() % screen_y) };
    }
}
void star_fall()
{
    int i;
    for (i = 0; i < scount; i++) {
        if (star[i].Y >= screen_y - 1) {
            star[i] = { SHORT(rand() % screen_x), 1 };
        }
        else {
            star[i] = { star[i].X, SHORT(star[i].Y + 1) };
        }
    }
}
void fill_star_to_buffer()
{
    for (int i = 0; i < scount; ++i) {
        consoleBuffer[star[i].X + screen_x * star[i].Y].Char.AsciiChar = '*';
        consoleBuffer[star[i].X + screen_x * star[i].Y].Attributes = 7;
    }
}
void draw_ship(int x, int y, int color)
{
    ship = { SHORT(x), SHORT(y) };
    consoleBuffer[ship.X + screen_x * ship.Y].Char.AsciiChar = '<';
    consoleBuffer[ship.X + 1 + screen_x * ship.Y].Char.AsciiChar = '0';
    consoleBuffer[ship.X + 2 + screen_x * ship.Y].Char.AsciiChar = '>';
    consoleBuffer[ship.X + screen_x * ship.Y].Attributes = color;
    consoleBuffer[ship.X + 1 + screen_x * ship.Y].Attributes = color;
    consoleBuffer[ship.X + 2 + screen_x * ship.Y].Attributes = color;
}
int main()
{
    bool play = true;
    DWORD numEvents = 0;
    DWORD numEventsRead = 0;
    srand(time(NULL));
    setConsole(screen_x, screen_y);
    setMode();
    init_star();

    while (play)
    {
        GetNumberOfConsoleInputEvents(rHnd, &numEvents);
        if (numEvents != 0)
        {
            INPUT_RECORD* eventBuffer = new INPUT_RECORD[numEvents];
            ReadConsoleInput(rHnd, eventBuffer, numEvents, &numEventsRead);
            for (DWORD i = 0; i < numEventsRead; ++i)
            {
                if (eventBuffer[i].EventType == KEY_EVENT &&

```

```

        eventBuffer[i].Event.KeyEvent.bKeyDown == true)
    {
        if (eventBuffer[i].Event.KeyEvent.wVirtualKeyCode
== VK_ESCAPE)
        {
            play = false;
        }
        else if
(eventBuffer[i].Event.KeyEvent.uChar.AsciiChar == 'c')
        {
            color = rand() % 255 + 1;
        }
    }
    else if (eventBuffer[i].EventType == MOUSE_EVENT)
    {
        int posX =
eventBuffer[i].Event.MouseEvent.dwMousePosition.X;
        int posY =
eventBuffer[i].Event.MouseEvent.dwMousePosition.Y;
        if (eventBuffer[i].Event.MouseEvent.dwButtonState &
FROM_LEFT_1ST_BUTTON_PRESSED)
        {
            color = rand() % 255 + 1;
        }
        else if
(eventBuffer[i].Event.MouseEvent.dwEventFlags & MOUSE_MOVED)
        {
            sposX = posX;
            sposY = posY;
        }
    }
    }
    delete[] eventBuffer;
}
star_fall();
for (int i = 0; i < scount; i++)
{
    if ((ship.X == star[i].X || star[i].X == ship.X + 1 || star[i].X
== ship.X + 2) && star[i].Y == ship.Y)
    {
        star[i] = { SHORT(rand() % screen_x), 1 };
        cnt++;
    }
    if (cnt >= 10)
    {
        play = false;
    }
}
clear_buffer();
fill_star_to_buffer();
draw_ship(sposX, sposY, color);
fill_buffer_to_console();
Sleep(200);
}
return 0;
}

```

