# Lab 8

## Assignment Program

```
#define _CRT_SECURE_NO_WARNINGS 1
#include <stdio.h>
#include <conio.h>
#include <windows.h>
#include <time.h>
#define scount 80
#define screen_x 80
#define screen_y 25

HANDLE wHnd;
HANDLE rHnd;
DWORD fdwMode;
DWORD numEvents = 0;
DWORD numEventsRead = 0;
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];
int x, y;
int X = 0, Y = 0;
int color = 9;
int hp = 10;

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

void draw_ship(int posx, int posy) {
    consoleBuffer[posx + screen_x * posy].Char.AsciiChar = '<';
    consoleBuffer[posx + screen_x * posy].Attributes = color;
    consoleBuffer[posx + 1 + screen_x * posy].Char.AsciiChar = '-';
    consoleBuffer[posx + 1 + screen_x * posy].Attributes = color;
    consoleBuffer[posx + 2 + screen_x * posy].Char.AsciiChar = '0';
    consoleBuffer[posx + 2 + screen_x * posy].Attributes = color;
    consoleBuffer[posx + 3 + screen_x * posy].Char.AsciiChar = '-';
    consoleBuffer[posx + 3 + screen_x * posy].Attributes = color;
    consoleBuffer[posx + 4 + screen_x * posy].Char.AsciiChar = '>';
```

```
    consoleBuffer[posx + 4 + screen_x * posy].Attributes = color;
}


int setConsole(int x, int y)
{
    wHnd = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleWindowInfo(wHnd, TRUE, &windowSize);
    SetConsoleScreenBufferSize(wHnd, bufferSize);
    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 = 1;
        }
    }
}
void init_star()
{
    for (int i = 0; i < 80; i++)
    {
        star[i].X = rand() % 80;
        star[i].Y = rand() % 25;
    }
}


void fill_star_to_buffer()
{
    for (int i = 0; i < 80; i++)
    {
        consoleBuffer[star[i].X + screen_x * star[i].Y].Char.AsciiChar = '*';
        consoleBuffer[star[i].X + screen_x * star[i].Y].Attributes = 7;
    }

}

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] = { short(star[i].X), short(star[i].Y + 1) };
            if (star[i].X >= x && star[i].X <= x + 4 && star[i].Y == y) {
                hp--;
                star[i] = { short(rand() % screen_x),1 };
            }
        }
    }
}
void fill_buffer_to_console()
{
    WriteConsoleOutputA(wHnd, consoleBuffer, bufferSize, characterPos,
&windowSize);
}

int main()
{
    int i;
    srand(time(NULL));
    bool play = true;
    setConsole(screen_x, screen_y);
    setMode();
    init_star();

    while (play && hp > 0)
    {
        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;
                    }
                    if (eventBuffer[i].Event.KeyEvent.uChar.AsciiChar == 'c')
                    {
                        color = rand() % 15 + 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() % 15 + 1;
                    }
                    else if (eventBuffer[i].Event.MouseEvent.dwEventFlags &
MOUSE_MOVED) {
                        x = posx - 2;
                        y = posy;
                    }
                }
            }
            delete[] eventBuffer;
        }
        i = 0;

        star_fall();
        clear_buffer();
        fill_star_to_buffer();
        draw_ship(x, y);
        fill_buffer_to_console();
        Sleep(150);
        i++;
    }
    return 0;
}
```