Assignment 4 : ให้นักศึกษาเพิ่มโปรแกรมจาก Assignment 3 ให้มีการสุ่มตำแหน่งการแสดงผล “*” บนจอภาพ ถ้ายานเคลื่อนที่ไปชน “*” จะมีการสุ่มเพื่อเปลี่ยนตำแหน่ง “*” และโปรแกรมจะหยุดการทำงานเมื่อมีการกดปุ่ม ESC หรือมีการชน “*” 10 ครั้ง

การส่งงาน : ให้นักศึกษาส่งเฉพาะ Assignment 4

```c
#include <stdio.h>
#include <windows.h>
#include <time.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];
int sky;
int john;
int din = 7;
int count = 0;
int setMode()
{
    rHnd = GetStdHandle(STD_INPUT_HANDLE);
    fdwMode = ENABLE_EXTENDED_FLAGS | ENABLE_WINDOW_INPUT |
        ENABLE_MOUSE_INPUT;
    SetConsoleMode(rHnd, fdwMode);
    return 0;
}
int setConsole(int x, int y)
{
    wHnd = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleWindowInfo(wHnd, TRUE, &windowSize);
    SetConsoleScreenBufferSize(wHnd, bufferSize);
    return 0;
}
void fill_data_to_buffer()
{
    for (int y = 0; y < screen_y; ++y) {
        for (int x = 0; x < screen_x; ++x) {
            consoleBuffer[x + screen_x * y].Char.AsciiChar = 'A' + rand() % 26;
            consoleBuffer[x + screen_x * y].Attributes = rand() % 255;
        }
    }
}
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 < 80; i++) {
        star[i].X = rand() % 80;
        star[i].Y = rand() % 25;
    }
}
void star_fall()
{
    int i;
    for (i = 0; i < scount; i++) {
        if (star[i].Y >= screen_y - 1) {
            star[i] = { (rand() % screen_x),1 };
        }
        else {
            star[i] = { star[i].X,star[i].Y + 1 };
        }
    }
}
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 fill_ship_to_buffer()
{
    consoleBuffer[sky  + screen_x * john].Char.AsciiChar = 'O';
    consoleBuffer[sky + screen_x * john].Attributes = din;
}
int chon() {
    for (int i = 0; i < 80; i++) {
        if (star[i].X == sky && star[i].Y == john) {
            count++;
            star[i].X = rand() % 80;
            star[i].Y = 1;
            if (count == 10) {
                return 1;
            }
        }
    }
    return 0;
}
int main()
{
    srand(time(NULL));
    init_star();
    bool play = true;
    DWORD numEvents = 0;
    DWORD numEventsRead = 0;
    setConsole(screen_x, screen_y);
    setMode();
    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;
                    }
                    if (eventBuffer[i].Event.KeyEvent.uChar.AsciiChar == 'c')
                        din = 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) {
                        din = rand() % 255 + 1;
                    }
                    else if (eventBuffer[i].Event.MouseEvent.dwEventFlags &
                        MOUSE_MOVED) {
                        sky = posx;
                        john = posy;
                    }
                }
            }
            delete[] eventBuffer;
        }
        star_fall();
        clear_buffer();
        fill_ship_to_buffer();
        fill_star_to_buffer();
        if (chon()) play = false;
        fill_buffer_to_console();
        Sleep(100);
    }
    return 0;
}
```