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;

}