Lab 7

Assignment Program

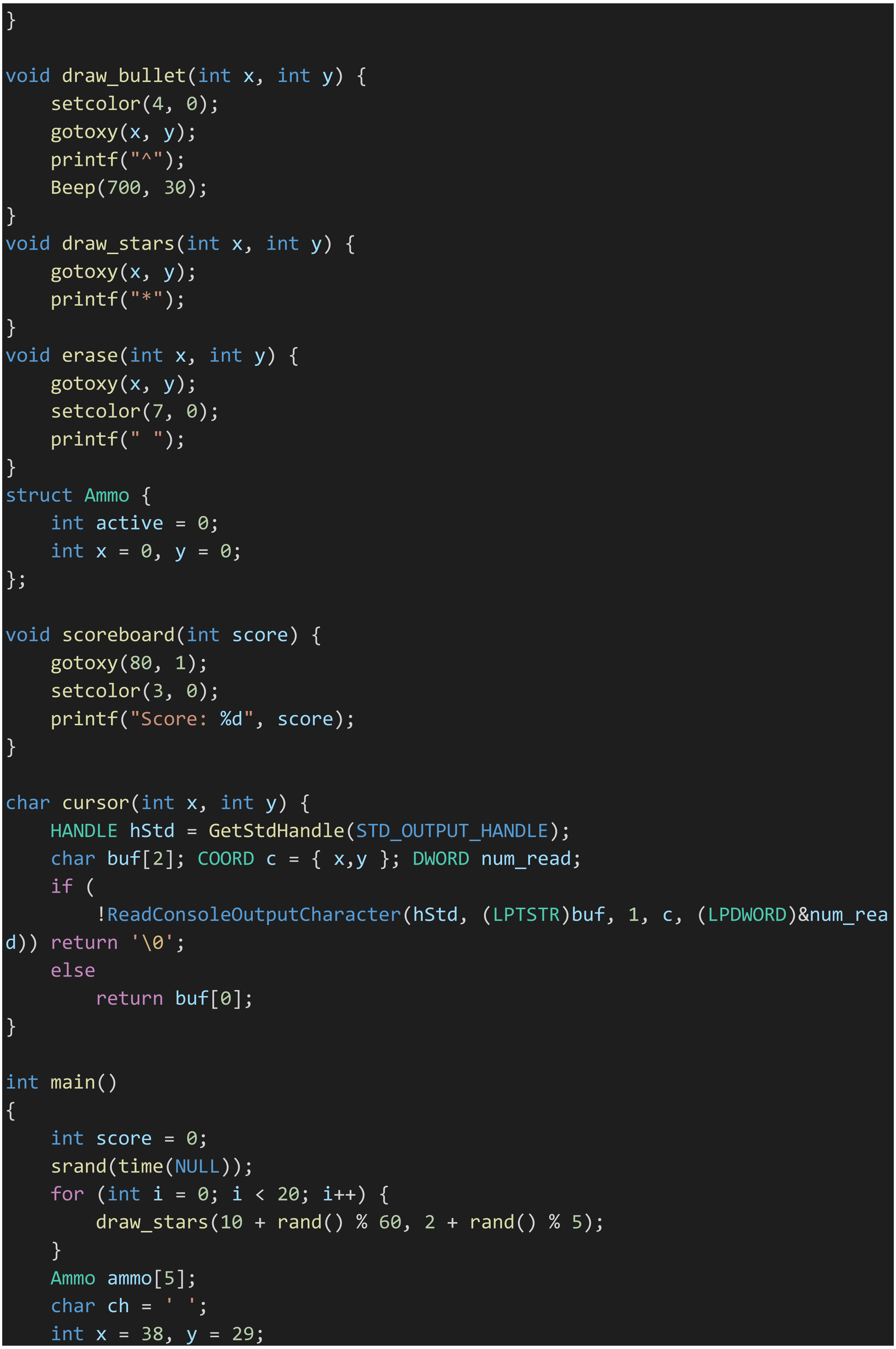
```
#define _CRT_SECURE_NO_WARNINGS 1
#include<stdio.h>
#include<conio.h>
#include<windows.h>
#include<time.h>
void setcolor(int fg, int bg)
{
    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hConsole, bg * 16 + fg);
}

void setcursor(bool visible)
{
    HANDLE console = GetStdHandle(STD_OUTPUT_HANDLE);
    CONSOLE_CURSOR_INFO lpCursor;
    lpCursor.bVisible = visible;
    lpCursor.dwSize = 20;
    SetConsoleCursorInfo(console, &lpCursor);
}


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

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

void draw_ship(int x, int y)
{
    setcolor(9, 1);
    gotoxy(x, y);
    printf(" <-0-> ");
```



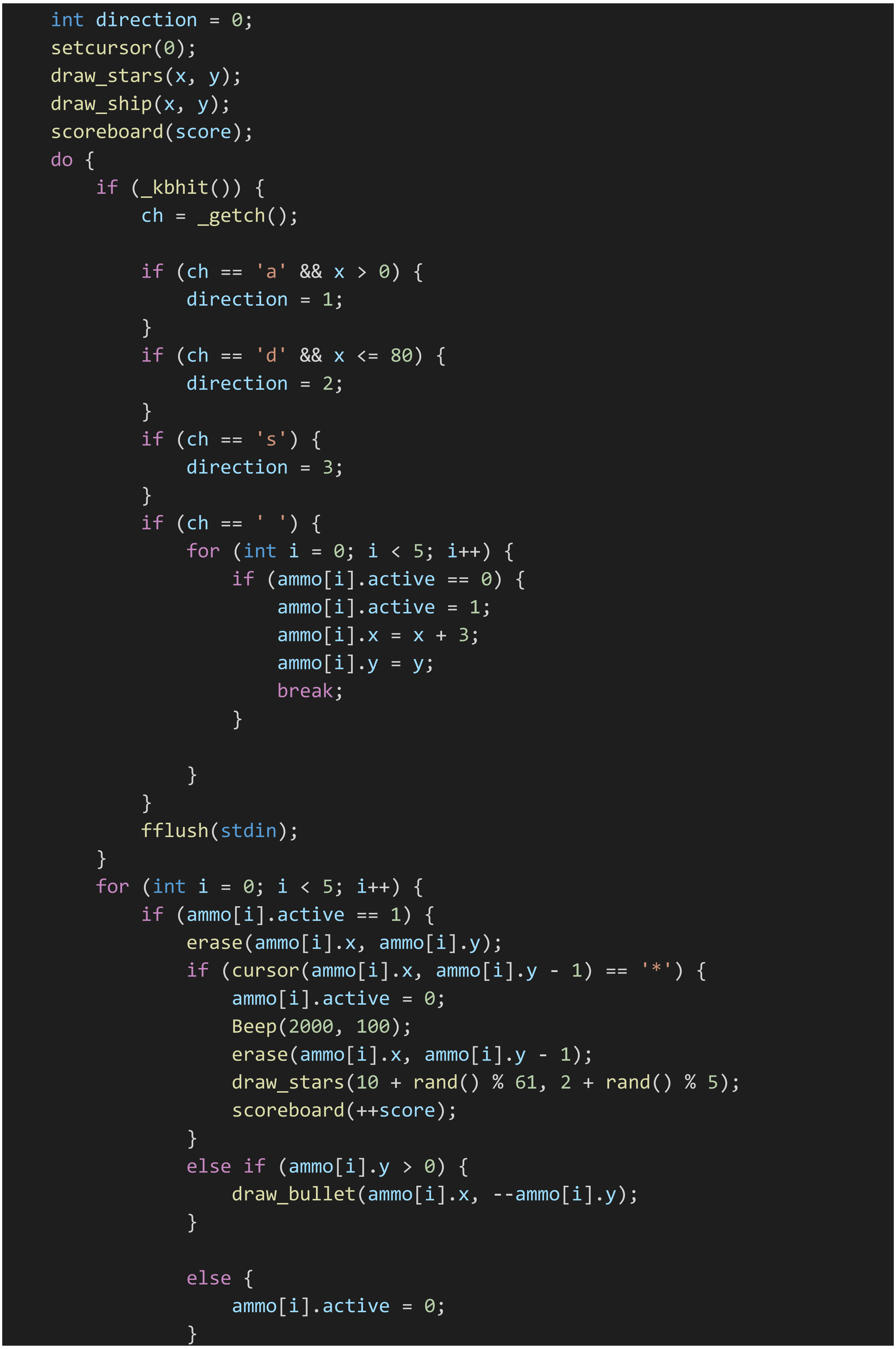
```
}

void draw_bullet(int x, int y) {
    setcolor(4, 0);
    gotoxy(x, y);
    printf("^");
    Beep(700, 30);
}
void draw_stars(int x, int y) {
    gotoxy(x, y);
    printf("*");
}
void erase(int x, int y) {
    gotoxy(x, y);
    setcolor(7, 0);
    printf(" ");
}
struct Ammo {
    int active = 0;
    int x = 0, y = 0;
};

void scoreboard(int score) {
    gotoxy(80, 1);
    setcolor(3, 0);
    printf("Score: %d", score);
}

char cursor(int x, int y) {
    HANDLE hStd = GetStdHandle(STD_OUTPUT_HANDLE);
    char buf[2]; COORD c = { x,y }; DWORD num_read;
    if (
        !ReadConsoleOutputCharacter(hStd, (LPTSTR)buf, 1, c, (LPDWORD)&num_rea
d)) return '\0';
    else
        return buf[0];
}

int main()
{
    int score = 0;
    srand(time(NULL));
    for (int i = 0; i < 20; i++) {
        draw_stars(10 + rand() % 60, 2 + rand() % 5);
    }
    Ammo ammo[5];
    char ch = ' ';
    int x = 38, y = 29;
```

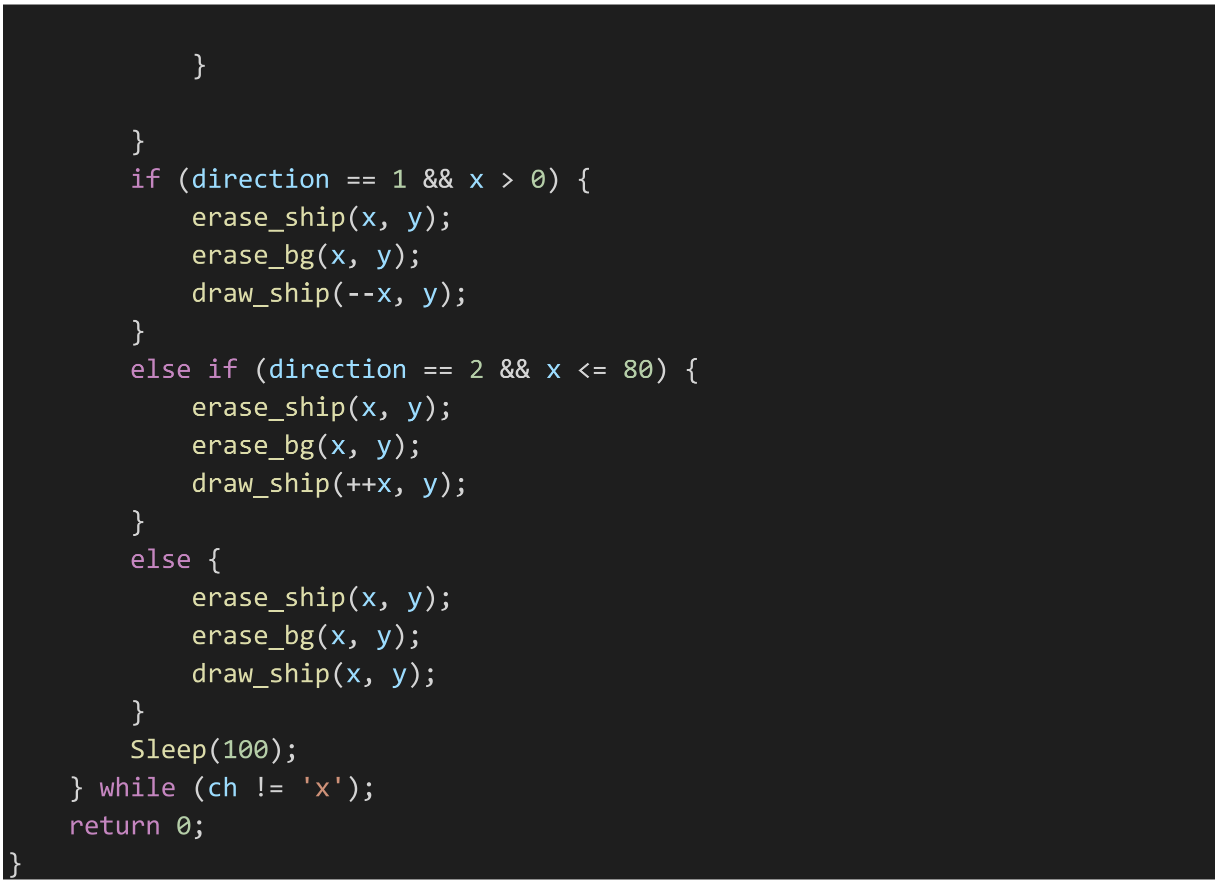


```
    int direction = 0;
    setcursor(0);
    draw_stars(x, y);
    draw_ship(x, y);
    scoreboard(score);
    do {
        if (_kbhit()) {
            ch = _getch();

            if (ch == 'a' && x > 0) {
                direction = 1;
            }
            if (ch == 'd' && x <= 80) {
                direction = 2;
            }
            if (ch == 's') {
                direction = 3;
            }
            if (ch == ' ') {
                for (int i = 0; i < 5; i++) {
                    if (ammo[i].active == 0) {
                        ammo[i].active = 1;
                        ammo[i].x = x + 3;
                        ammo[i].y = y;
                        break;
                    }

                }
            }
            fflush(stdin);
        }
        for (int i = 0; i < 5; i++) {
            if (ammo[i].active == 1) {
                erase(ammo[i].x, ammo[i].y);
                if (cursor(ammo[i].x, ammo[i].y - 1) == '*') {
                    ammo[i].active = 0;
                    Beep(2000, 100);
                    erase(ammo[i].x, ammo[i].y - 1);
                    draw_stars(10 + rand() % 61, 2 + rand() % 5);
                    scoreboard(++score);
                }
                else if (ammo[i].y > 0) {
                    draw_bullet(ammo[i].x, --ammo[i].y);
                }

                else {
                    ammo[i].active = 0;
                }
```



```
            }

        }
        if (direction == 1 && x > 0) {
            erase_ship(x, y);
            erase_bg(x, y);
            draw_ship(--x, y);
        }
        else if (direction == 2 && x <= 80) {
            erase_ship(x, y);
            erase_bg(x, y);
            draw_ship(++x, y);
        }
        else {
            erase_ship(x, y);
            erase_bg(x, y);
            draw_ship(x, y);
        }
        Sleep(100);
    } while (ch != 'x');
    return 0;
}
```