

## Commie Invaders

The Communists are attacking and this time they are coming from space! Luckily you have the ReaganX anti-commie laser. You must defend Freedom!!! Use the arrow keys to move left and right and press the space key to fire. As you destroy the coming invaders you will be granted more lives, and you'll need them because they will get faster as you get closer to the end. To win you need to shoot down 100 commie invaders.

Commie Invaders is written by James E. Ward.

```
// commieinvaders.c listing begins:

/* Commie Invaders by James E. Ward 21 December 2008 you may edit, distribute
 *without cost or copy this file as long as the original author is credited.
 *Special thanks to Joe Larson for helping me out with the timing.*/

#include <curses.h>
#include <time.h>
#include <stdlib.h>
#include <ctype.h>
/*dont change these two*/
#define Y 0
#define X 1
/*you can change these 3 to effect difficulty and to fit your console*/
#define NUM_COM 8
#define SIZE_X 61
#define SIZE_Y 24

float magicNumber= CLOCKS_PER_SEC;

char grid[SIZE_Y][SIZE_X];

void init () {
    srand (time(NULL));
    keypad(initscr(),1);
    raw ();
    nodelay(stdscr,0);
    noecho();
    curs_set(0);
    nonl();
}

void initgrid() {
    for (int yy=0;yy<SIZE_Y;yy++) {
        for (int xx=0;xx<SIZE_X;xx++) {
            grid[yy][xx]=(yy==0 || yy== SIZE_Y-1 || xx==0 || xx==SIZE_X-1)? '#' : ' ';
        }
    }
}

void play(void) {
    int input,score=0,lives=5,com[NUM_COM][2], playerx=SIZE_X /2;
    int playery=SIZE_Y - 2, turn=0, last,level=1, lv=0;
    char comsym[NUM_COM];

    nodelay(stdscr,1);
    for (int i = 0; i<NUM_COM;i++) {
        com[i][X]=0;
        com[i][Y]=0;
    }
    last = clock ();
    do {/*main game loop*/

        magicNumber = CLOCKS_PER_SEC / ((float)(level - 0) / 2);

        for (int i=0;i<NUM_COM;i++) {
            comsym[i]=(com[i][Y]==0) ? '#' : '0';
        }

        /*Draw the screen*/
        for (int yy=0;yy<SIZE_Y;yy++) {
```

// Listing continued on next page...

// Listing continued from previous page

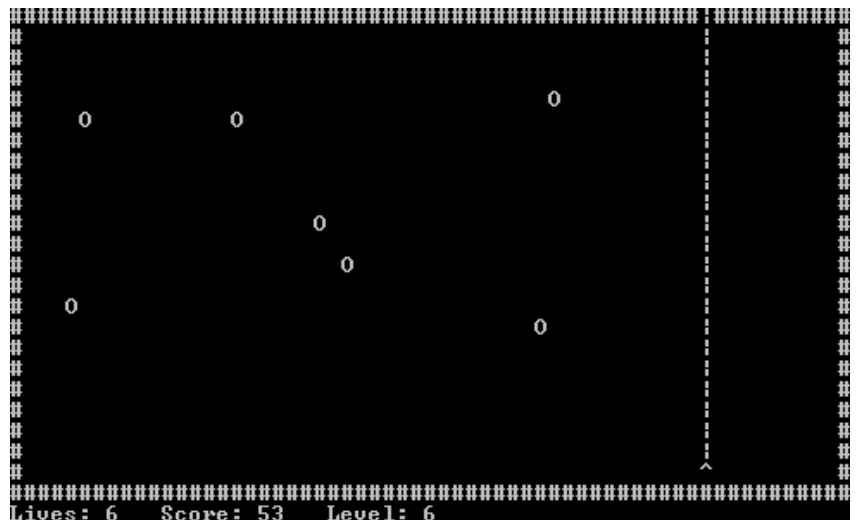
```

    for (int xx=0;xx<SIZE_X;xx++) {
        mvaddch(yy,xx,grid[yy][xx]);
    }
}
/*Draw player and score bar*/
mvaddch(playery,playerx,'^');
mvprintw(SIZE_Y,0,"Lives: %d   Score: %d   Level: %d   ",lives,score,level);
/*draw communists*/
for (int i = 0;i<NUM_COM;i++) {
    mvaddch(com[i][Y],com[i][X],comsym[i]);
}
refresh();

/*Get input*/
if (KEY_LEFT == input && playerx>1) {
    playerx--;
}
if (KEY_RIGHT == input && playerx<(SIZE_X-2)) {
    playerx++;
}
if ('p' == input) {
    mvprintw(SIZE_Y,50,"paused");
    while ('p'!=(input=getch())) {}mvprintw(SIZE_Y,50,"");
}
if (' ' ==input || KEY_UP ==input) {
    int laser=playery-1;
    for (;laser>=0;laser--) {
        mvaddch(laser,playerx,'|');
    }
    refresh();
    napms(50);
    for (int i = 0;i<NUM_COM;i++) {
        if (com[i][X]== playerx&& com[i][Y]!=0) {
            score++;
            com[i][Y]=0;
            beep();
            lv++;
            if (lv==10){level++;lv=0;}
            if (score==25||score==50||score==75) {
                lives++;
            }
        }
    }
}
}

turn = clock() - last;
/*Fun with Commies*/
if (turn>=magicNumber) {
    int rn;
    last = clock ();
    for (int i = 0;i<NUM_COM;i++) {
        if (com[i][Y]==0) {
            rn= rand()%4;
            if (rn==1) {
                com[i][Y]=1;
                com[i][X]=rand() % (SIZE_X-2)+1;
            }
        } else {
            rn= rand()%5;
            switch (rn) {
                case 0:
                    com[i][Y]++;
                    break;

```



// Listing continued on next page...

// Listing continued from previous page

```
        case 1:
            com[i][Y]++;
            break;
        case 2:
            com[i][X]+=(com[i][X]<SIZE_X-2)?1:0;
            break;
        case 3:
            com[i][X]-=(com[i][X]>1)?1:0;
            break;
        case 4:
            com[i][Y]++;
            break;
        default :
            com[i][Y]++;
            break;
    }
}
if (com[i][Y]==playery) {
    lives--;
    com[i][Y]=0;
}
}

turn=00;
}
} while ('x'!=(input=getch()) && lives >= 0 && score < 100);

clear();
mvprintw(0,0,"You killed %d Commies",score);
if (lives<0) {
    mvprintw(1,0,"then died!");
}
if (score>99) {
    mvprintw(1,0,"You are a heroe of the free world!");
}
nodelay(stdscr,0);
}

int main(int argc,char* argv[]) {

    init();
    mvprintw(1,0,"The Communists are attacking and this time they are coming");
    mvprintw(2,0,"from space! Luckily you have the ReaganX anti-commie laser.");
    mvprintw(3,0,"You must defend Freedom!!!");
    getch();
    initgrid();
    play();
    char again='+';
    while (1) {
        mvprintw(3,0,"Play again? (y/n)");
        if ('y'==(again=getch())|| again=='Y') {
            play();
        }
        if (again=='n' || again == 'N') {
            break;
        }
    }
    exit(endwin());
}
```