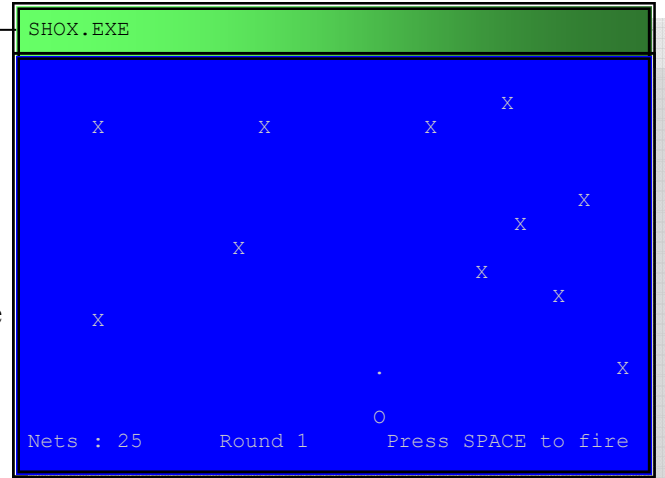


Shox

Checking into work at Shox Biotech/Robotic Engineering Labs you discover the mutated creatures have escaped their cages and are making their way out of the building. Not 100% prepared for such a task you have only limited capture nets, but you only have to hold the tide of creatures back until reinforcements arrive (which is in approximately 10 levels). You set up a perimeter and deploy the containment protocol bot which runs around the perimeter and shoots capture nets at the creatures. You initially can only find slightly more ammo than the bot needs, but are sure you can find more and refill it as it moves from one room to another.

Shox is a game of resource management, your resources being your ammunition. Run through your nets too quickly and you'll find yourself struggling through later rounds where higher speeds mean more frequent misses. But if you can leave yourself with enough nets in the end aiming could be unnecessary as you can, if you plan it right, simply fire off enough to cover every line. You save yourself nets by, first of all not missing and second, taking the opportunity to catch two or more with one net by spotting where they line up and netting them from the right direction. Good luck!

Shox is written by Joseph Larson.



SHOX.C	You will need: a C/C++ compiler with curses.
<pre>#include <curses.h> #include <time.h> #include <stdlib.h> #include <ctype.h> int dirs[6] = {0, 1, 0, -1, 0, 1}; void init () { srand (time(NULL)); initscr (); raw (); nodelay(stdscr,1); noecho(); curs_set(0); nonl(); keypad(stdscr,1); } void play () { int x, y, bx, by, c, d, in, ammo, lvl, buf[LINES][COLS]; y = x = d = lvl = 0; ammo = 10; /* Set up variables for new game. */ do { /* Set up variables for new level */ clear (); lvl++; ammo += 15; mvprintw (LINES - 1, 37, "Round %d", lvl); for (by = 0; by < LINES; by++) for (bx = 0; bx < COLS; bx++) buf[by][bx] = 0; /* clear buffer */ for (c = 0; c < 15; c++) { /* Set up enemies. */ do { by = 1 + rand () % (LINES - 3); bx = 1 + rand () % (COLS - 2); } while (buf[by][bx]); buf[by][bx] = 1; mvaddch (by, bx, 'X'); } /* c will count targets */ /* Draw last few more things on the screen. */ mvaddch (y, x, 'O'); mvprintw (LINES - 1, 0, "Nets : %d ", ammo); /* space after the d */ mvprintw (LINES - 1, 60, "Press SPACE to fire."); } }</pre>	

Listing continued on page 2...

SHOX.C	Listing Continued from page 1....
<pre> refresh (); while ((ammo > 0) && (c > 0)) { /* The main game loop */ in = getch (); if (tolower(in) == 'q') return; if (in == ' ') { /* shoot a bullet */ ammo--; mvprintw (LINES - 1, 0, "Nets : %d ", ammo); /* space after again. */ bx = x; by = y; do { bx += dirs[d + 1]; by += dirs[d + 2]; /* move the bullet */ if (buf[by][bx]) { /* a hit */ beep (); buf[by][bx] = 0; c--; } mvaddch (by, bx, '.'); refresh (); napms (15); mvaddch (by, bx, ' '); } while (!((bx==0) + (by==0) + (bx==COLS - 1) + (by==LINES - 2))); /* above line is for edge detection. */ } mvaddch (y, x, ' '); x += dirs[d]; y += dirs[d + 1]; if (((x == 0) + (y == 0) + (x == COLS - 1) + (y == LINES - 2)) == 2) { /* if we're at a corner change direction. */ d++; d %= 4; } mvaddch (y, x, 'O'); refresh (); napms (275 - 25 * lvl); } /* end of level loop if out of bullets or out of enemies. */ } while ((c == 0) && (lvl < 10)); if (c) mvprintw (10, 27, "Out of bullets. Game over."); else mvprintw (10, 15, "Congratulations! You have defeated every enemy!"); refresh (); } int main () { init (); play (); mvprintw (15, 32, "Press any key..."); nodelay(stdscr,0); getch (); endwin (); return 0; } </pre>	

Author's Notes:

Small, extremely simple to play, Shox still manages to inspire an almost instant desire to see it do more. I've had suggestions for blocks that can't be shot through, different enemies and different ammunition types for super enemies that are immune to the normal shot. Awesome suggestions, all of them, but none of them were implemented in this version.

As it stands this program is, in my opinion, a perfectly simple example for the curses library. As such any additions to it, in it's current form, would only complicate matters and make the code unnecessarily longer.

Still, Shox 2 could be a lot of fun.

If you find the game becoming too easy you can change the initial amount of the ammo variable. Try setting it to zero and you'll have only as many nets carry over into the next round as you save in this one by hitting 2 with one.