Fish Hunt

This is a fishing game where you are the fisherman. Simple enough. Just give your location on the lake and how far down to drop your hook, and your state of the art radar fish locator will tell you where you need to go.

Functionally, this is just a number guessing game in 3 dimensions with a "stars" like distance indicator. However, it's use of text as graphics make it easier to look at. Some may find the way this game gives hints practically gives away the location of the fist, particularly in a such a small pond. It should be no challenge at all, even for the novice programmer, to rewrite this game to make your search area larger and the hints more cryptic.

Fish Hunt is by Glen Williams, inspired on the BASIC game "Stupid Fish" from 'The Rainbow Book of BASIC Program' © Grisewood & Dempsey Ltd 1984.

/* fishhunt.c listing begins: */

```
/* ============================================
 * == This is a game called Fish-Hunt, where you
 * == have to locate and catch the fish.
 * == Original (Stupid Fish) is from a book called
 * == The Rainbow Book Of Basic Programs.
 * == This is my C conversion of the game.
 * == By Glen Williams
 * == Created 31/05/2009
 * == Amended 07/06/2009
 * == Notes : none
 * ============================================
 */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>                    /* For random seed */

/* == Constants */
#define TRUE 1
#define FALSE 0
#define BOXSIZE 8                    /* Determines the size of our sea/box.*/

/* Functions */
void drawHeader(void);               /* Main menu */
void drawSubHeader(void);            /* Sub menu */
void drawEndOfGame();                /* End Game screen */
void clearScreen(int numToClear);    /* Cheap screen clearer */
void sonarReport(int boatPos, int fishPos); /* Draw a line of stars */
int getRandomNumber(int upTo);       /* Random number generator */

int main()
{
    time_t t;                        /* Used for random seed */

    /* Genrate random seed, make sure this goes before all the other */
    srand(time(&t));

    int fishX = 0;                   /* Holds the fish X (EAST/WEST) postion */
    int fishDepth = 0;               /* Holds the fish depth of the fish */
    int fishZ = 0;                   /* Holds the fish Z (NORTH/SOUTH) postion */
    int boatX = 0;                   /* Holds boat X (EAST/WEST) postion */
    int netDepth = 0;                /* Holds net depth */
    int boatZ = 0;                   /* Holds the boat Z (NORTH/SOUTH) postion */
    int turnsTaken = 0;   /* Holds the amount of moves taken to find the fish */
    int win = FALSE;                 /* Holds the game win state */

    /* Hide the fish/catch :) */
    fishX = getRandomNumber(BOXSIZE);
    fishDepth = getRandomNumber(BOXSIZE);
    fishZ = getRandomNumber(BOXSIZE);

    /* For testing only, show the fish location */
    /*printf("fishN/S=%d, fishE/W=%d, fishDepth=%d\n",fishZ,fishX,fishDepth); */

    drawHeader();
    drawSubHeader();

    do {
    /* Get user input */
    printf("\nEnter your NORTH/SOUTH position : ");
    scanf("%d",&boatZ);
    printf("Enter your WEST/EAST postion : ");
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
    scanf("%d",&boatX);
    printf("Enter your net depth : ");
    scanf("%d",&netDepth);

    turnsTaken++;

    /* Check for a win */
    if(boatX == fishX && netDepth == fishDepth && boatZ == fishZ)
        {
         win = TRUE;
         clearScreen(10);
         printf("\nWell done! You took %d turns to find the fish.\n",turnsTaken);
         drawEndOfGame();
        }
    else /* If no win found, try again */
        {
        clearScreen(18);
        drawHeader();
        printf("Turns taken = :     %d\n",turnsTaken);
        printf("=====================================\n");
        printf("Sorry, no fish found, please try again.\n");
        printf("Soner : More *** = closer\n\n");

        /* Compare boat position to fish position and report on it */

        /* NORTH/SOUTH */
        if(boatZ < fishZ)
            {
            printf("Sonar Report : Fish located more NORTH : ");
            sonarReport(boatZ, fishZ);
            }
        else if(boatZ > fishZ)
            {
            printf("Sonar Report : Fish located more SOUTH : ");
            sonarReport(boatZ, fishZ);
            }
        else if(boatZ == fishZ)
            {
            printf("NORTH/SOUTH postion found at           : %d\n",boatZ);
            }

        /* WEST/EAST */
        if(boatX < fishX)
            {
            printf("Sonar Report : Fish located more EAST  : ");
            sonarReport(boatX, fishX);
            }
        else if(boatX > fishX)
            {
            printf("Sonar Report : Fish located more WEST  : ");
            sonarReport(boatX, fishX);
            }
        else if(boatX == fishX)
            {
            printf("WEST/EAST postion found at             : %d\n",boatX);
            }

        /* NET DEPTH */
        if(netDepth < fishDepth)
            {
            printf("Sonar Report : Fish located DEEPER     : ");
            sonarReport(netDepth, fishDepth);
            }
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
        else if(netDepth > fishDepth)
            {
            printf("Sonar Report : Fish located SHALLOWER  : ");
            sonarReport(netDepth, fishDepth);
            }
        else if(netDepth == fishDepth)
            {
            printf("DEPTH postion found at                 : %d\n",netDepth);
            }
        }

    }while (win == FALSE);

    getchar();
    return EXIT_SUCCESS;
}

/* Basic game header logo */
void drawHeader()
{
    printf("======================================\n");
    printf("==       >o        >o             ==\n");
    printf("==     >o              >o    >o    ==\n");
    printf("== >o    >o   Fish-Hunt            ==\n");
    printf("==                          >o      ==\n");
    printf("==           >o        >o          ==\n");
    printf("======================================\n");
    clearScreen(3);
}
/* Game discription and instruction */
void drawSubHeader()
{
    printf("This is a fishing game where you the fisherman, are trying to "
      "catch your tea.\n");
    printf("The fish are located in a vast sea in which you must search and "
      "find your catch.\n");
    printf("Coordinates are as follows : -\n");
    printf("South starts at 0, and ends at North %d\n",BOXSIZE);
    printf("West starts at 0, and ends at East %d\n",BOXSIZE);
    printf("Depth starts at 0 (shallow) and ends at %d (deep).\n",BOXSIZE);
    printf("Note, keep track of your progress with a pencil and notepad.\n");
    printf("The sooner you find the fish, the sooner you can eat your tea :)"
      "\n\n\n");
}
/* Fish found/end of game screen */
void drawEndOfGame()
{
    printf("You can now cook your tea.\n\n");
    printf("       *****          *\n");
    printf("     **      **      *\n");
    printf("   **  0      *******\n");
    printf(" **               ***      Enjoy! :)\n");
    printf("   **  __-    *******\n");
    printf("     **      **      *\n");
    printf("       *****          *\n\n");
}
/* Basic screen clearing routine */
void clearScreen(int numToClear)
{
        int ctr = 0;

        for(ctr=0;ctr<numToClear;ctr++)
            {
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
            putchar('\n');
            }
}
/* Random number generator */
int getRandomNumber(int upTo)
{
    int result = 0;

    result = (rand() % upTo); /* Create a random number from 0 to x */

    return(result);
}
/* Draw the sonar progress bar */
void sonarReport(int boatPos, int fishPos)
{
    float d;

    for(d = BOXSIZE; (int)d > abs(boatPos - fishPos); d /=2)
    {
    printf("***");
    }
    printf("\n");
}
```