Under & Over

Under & Over is a dice game where players choose a number from 1 to 12, a dice is rolled, and you are rewarded if the number you choose fell in the same range above or below 7 as the roll, and awarded even more if you get it right on.

The rules are based on the BASIC game which originally introduced the author to Under & Over and do not represent the way the game would actually be played in a gambling establishment. In this version an elementary knowledge of the statistics of dice rolls will yield a strategy guaranteed to lead to a growing bankroll for the player.

As a challenge to the reader, rewrite the game so to introduce a house edge. In betting houses the players can only be on three options: over or under 7 paying out at even odds or exactly 7 paying out at 5 to 1. However, if you'd like to keep the aspect of paying out for guessing any number, try to adjust the payout.

Under & Over is by R. Alan Monroe inspired by a BASIC game of the same name from 'More BASIC Computer Games' ©1979 edited by David H. Ahl.

```
/* underover.c listing begins: */

//by R. Alan Monroe. Inspired by the BASIC game
#include <stdlib.h>
#include <curses.h>

const int chunk = RAND_MAX / 6;
const char *diespots[6][3] = { {"   ", " o ", "   "}, {"  o", "   ", "o  "},
  {"o  ", " o ", "  o"}, {"o o", "   ", "o o"}, {"o o", " o ", "o o"},
  {"o o", "o o", "o o"}};
int wide, high;

void printhelp() {
    mvprintw(1, (wide/2) - 11, "-=== UNDER & OVER ==--");
    mvprintw( (high-2), (wide/2) - 38,
"LEFT/RIGHT: Change Number - UP/DOWN: Adjust Bet - ENTER: Roll dice - Q: quit");
}

void clearhelp() {
    move( (high-2), 0);
    clrtobot();
}

void printnums(n) {
    int i;
    for (i=2; i<=12; i++)
        mvprintw( (high/5), ((wide*0.33)-18) + 4*(i-2), "%2d ", i);
    attron(A_REVERSE);
    mvprintw( (high/5), ((wide*0.33)-18) + 4*(n-2), "%2d ", n);
    attroff(A_REVERSE);
}

void printfunds(cash, maxcash, bet) {
    mvprintw( (high/2.7),   (wide*0.68), "BEST: $%8d", maxcash);
    mvprintw( (high/2.7)+1, (wide*0.68), "CASH: $%8d", cash);
    mvprintw( (high/2.7)+2, (wide*0.68), "BET:  $%8d", bet);
}

void printnotice(notice) {
    mvprintw( (high/2.7)+4, (wide*0.68), notice);
    clrtoeol();
}

void printdice(d1, d2) {
    int diesum = d1 + d2;
    mvprintw( (high/5)+2, ((wide*0.33)-18) ,
      "                                          "); // 34 spaces
    mvprintw( (high/5)+3, ((wide*0.33)-18) ,
      "                                          "); // 34 spaces
    mvprintw( (high/5)+4, ((wide*0.33)-18) ,
      "                                          "); // 34 spaces
    mvprintw( (high/5)+6, ((wide*0.33)-18) ,
      "                                          "); // 34 spaces
    mvprintw( (high/5)+7, ((wide*0.33)-18) ,
      "                                          "); // 34 spaces
    mvprintw( (high/5)+8, ((wide*0.33)-18) ,
      "                                          "); // 34 spaces
    attron(A_REVERSE);
    mvprintw((high/5)+2, ((wide*0.33)-18)+(diesum-2)*4, "%s", diespots[d1-1][0]);
    mvprintw((high/5)+3, ((wide*0.33)-18)+(diesum-2)*4, "%s", diespots[d1-1][1]);
    mvprintw((high/5)+4, ((wide*0.33)-18)+(diesum-2)*4, "%s", diespots[d1-1][2]);
    mvprintw((high/5)+6, ((wide*0.33)-18)+(diesum-2)*4, "%s", diespots[d2-1][0]);
    mvprintw((high/5)+7, ((wide*0.33)-18)+(diesum-2)*4, "%s", diespots[d2-1][1]);
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
    mvprintw((high/5)+8, ((wide*0.33)-18)+(diesum-2)*4, "%s", diespots[d2-1][2]);
    attroff(A_REVERSE);
}

void printintro() {
    mvprintw( 1,               (wide/2) - 11, "--== UNDER & OVER ==--");
    mvprintw( (high * 0.25),   (wide/2) - 17,
      "This is a game of UNDER and OVER.");
    mvprintw( (high * 0.25)+1, (wide/2) - 35,
      "If you pick under 7 and the dice are also under 7, you win EVEN MONEY.");
    mvprintw( (high * 0.25)+2, (wide/2) - 34,
      "If you pick over 7 and the dice are also over 7, you win EVEN MONEY.");
    mvprintw( (high * 0.25)+3, (wide/2) - 29,
      "If your pick is SPOT ON you will win money at FOUR TO ONE.");
    mvprintw( (high * 0.25)+4, (wide/2) - 20,
      "You have $100 to start with. Good luck!");
    mvprintw( (high * 0.25)+6, (wide/2) - 13,
      "[Press any key to continue]");
    mvprintw( (high * 0.8)  , (wide/2) - 24,
      "Original BASIC version: Creative Computing, (c) 1979");
    mvprintw( (high * 0.8)+1, (wide/2) - 26,
      "C conversion for Cymon's Games: R. Alan Monroe, 2009");
    refresh();
    getch();
    clear();
}

void printoutro(cash) {
    if (cash > 0) {
        mvprintw( (high*0.8),   (wide/2)-13, "GOODBYE. ENJOY YOUR $%d!!!", cash);
    } else {
        mvprintw( (high*0.8),   (wide/2)-27,
          "THE GAME IS OVER AND YOU ARE FLAT BROKE. SORRY CHARLIE.");
        mvprintw( (high*0.8)+1, (wide/2)-21,
          "THIS PROVES THAT IS IS NOT GOOD TO GAMBLE.");
    }
    mvprintw( (high*0.8)+3, (wide/2)-11, "[Press any key to exit]");
    refresh();
    getch();
}

int main() {
    int k, die1, die2, dice = 0;
    int choice = 7;
    int cash = 100, maxcash = 100;
    int bet = 5;
    bool playing = TRUE;

    srand (time(NULL));
    initscr();
    clear(); noecho(); cbreak(); keypad(stdscr, TRUE); curs_set(0);

    getmaxyx(stdscr, high, wide);
    printintro();

    printnums(choice);
    printfunds(cash, maxcash, bet);
    printhelp();
    refresh();

    while (playing && cash > 0) {
        k = getch();
        switch (k) {
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
        case KEY_LEFT:
            if (choice > 2) choice--;
            printnums(choice);
            printnotice("");
            break;

        case KEY_RIGHT:
            if (choice < 12) choice++;
            printnums(choice);
            printnotice("");
            break;

        case KEY_UP:
            if (bet < cash) bet = bet + 5;
            printfunds(cash, maxcash, bet);
            printnotice("");
            break;

        case KEY_DOWN:
            if (bet > 5) bet = bet - 5;
            printfunds(cash, maxcash, bet);
            printnotice("");
            break;

        case 10:
        case KEY_ENTER:
            die1 = 1 + (rand() / chunk); if (die1 == 7) die1 = 6;
            die2 = 1 + (rand() / chunk); if (die2 == 7) die1 = 6;
            dice = die1+die2;
            printdice(die1, die2);
            if (dice==choice) {
                cash = cash + (4 * bet);
                if (cash > maxcash) maxcash = cash;
                printnotice("** YOU WIN 4 TO 1 **");
                printfunds(cash, maxcash, bet);
            } else if ((dice < 7 && choice < 7) || (dice > 7 && choice > 7)) {
                cash = cash + bet;
                if (cash > maxcash) maxcash = cash;
                printnotice("** YOU WIN EVEN MONEY **");
                printfunds(cash, maxcash, bet);
            } else if ((dice <= 7 && choice >= 7) || (dice >= 7 && choice<= 7)) {
                if (dice != 7) cash = cash - bet;
                if (bet > cash) bet = cash;
                printnotice("!! YOU LOSE !!");
                printfunds(cash, maxcash, bet);
            }
            break;

        case 'Q':
        case 'q':
            playing = FALSE;
            break;
        }
        refresh();
    }
    clearhelp();
    printoutro(cash);

    curs_set(1);
    endwin();
    return 0;
}
```

/* Listing continued on next page…*/