Chuck-a-luck

Chuck-a-luck is a game of chance where 3 dice are rolled and the player bets on what number they think the dice will roll too. If one or more dice roll to their number the player wins for each dice that lands. The game is easy to pick up, easy to play, but hard to win at.

This version of the game doesn't include all the betting options that a player could encounter were they to meet this game in real life. There is also the option of placing a bet on a triple (all three dice landing any one number) with 30 to 1 payout odds if you win. There are also sometimes "Big" offered where the player wins if the total of all the dice is 11 or higher and not a triple or "Smalls" where the player wins if the total is 9 or less and not a triple or "Fields" where the player wins if the total falls outside the range of 8 to 12, all paying out to even money. A challenge to the reader would be to add these betting options in as well as allow multiple bets.

Chuck-a-luck by R. Alan Monroe inspired by a BASIC game of the same name from 'More BASIC Computer Games' (c)1979 edited by David H. Ahl..

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

#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, "-=== CHUCK-A-LUCK ==--");
    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=1; i<=6; i++) {
        mvprintw( (high/5), ((wide*0.33)-9) + 4*(i-2), "%2d ", i);
    }
    attron(A_REVERSE);
    mvprintw( (high/5), ((wide*0.33)-9) + 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, d3) {
    mvprintw( (high/5)+2, ((wide*0.33)-13) , "                          ");
    mvprintw( (high/5)+3, ((wide*0.33)-13) , "                          ");
    mvprintw( (high/5)+4, ((wide*0.33)-13) , "                          ");
    mvprintw( (high/5)+6, ((wide*0.33)-13) , "                          ");
    mvprintw( (high/5)+7, ((wide*0.33)-13) , "                          ");
    mvprintw( (high/5)+8, ((wide*0.33)-13) , "                          ");
    mvprintw( (high/5)+10, ((wide*0.33)-13) , "                          ");
    mvprintw( (high/5)+11, ((wide*0.33)-13) , "                          ");
    mvprintw( (high/5)+12, ((wide*0.33)-13) , "                          ");

    attron(A_REVERSE);
    mvprintw( (high/5)+2, ((wide*0.33)-9) + (d1-2)*4, "%s", diespots[d1-1][0]);
    mvprintw( (high/5)+3, ((wide*0.33)-9) + (d1-2)*4, "%s", diespots[d1-1][1]);
    mvprintw( (high/5)+4, ((wide*0.33)-9) + (d1-2)*4, "%s", diespots[d1-1][2]);
    mvprintw( (high/5)+6, ((wide*0.33)-9) + (d2-2)*4, "%s", diespots[d2-1][0]);
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
    mvprintw( (high/5)+7, ((wide*0.33)-9) + (d2-2)*4, "%s", diespots[d2-1][1]);
    mvprintw( (high/5)+8, ((wide*0.33)-9) + (d2-2)*4, "%s", diespots[d2-1][2]);
    mvprintw( (high/5)+10, ((wide*0.33)-9) + (d3-2)*4, "%s", diespots[d3-1][0]);
    mvprintw( (high/5)+11, ((wide*0.33)-9) + (d3-2)*4, "%s", diespots[d3-1][1]);
    mvprintw( (high/5)+12, ((wide*0.33)-9) + (d3-2)*4, "%s", diespots[d3-1][2]);
    attroff(A_REVERSE);
}

void printintro() {
    mvprintw( 1,               (wide/2) - 11, "--== CHUCK-A-LUCK ==--");
    mvprintw( (high * 0.25),   (wide/2) - 24,
        "Choose a number from 1 to 6. I will roll 3 dice.");
    mvprintw( (high * 0.25)+1, (wide/2) - 25,
        "If your number matches 1 die, I pay off EVEN MONEY.");
    mvprintw( (high * 0.25)+2, (wide/2) - 27,
        "If your number matches 2 dice, I pay off at TWO TO ONE.");
    mvprintw( (high * 0.25)+3, (wide/2) - 28,
        "If your number matches 3 dice, I pay off at THREE 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, 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 IT 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, die3, matches;
    int choice = 1;
    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();
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

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

    while (playing && cash > 0) {
        k = getch();
        switch (k) {
        case KEY_LEFT:
            if (choice > 1) choice--;
            printnums(choice);
            printnotice("");
            break;

        case KEY_RIGHT:
            if (choice < 6) 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;
            die3 = 1 + (rand() / chunk); if (die3 == 7) die1 = 6;
            printdice(die1, die2, die3);
            matches = 0;
            if (choice==die1) matches++;
            if (choice==die2) matches++;
            if (choice==die3) matches++;
            switch (matches) {
                case 1:
                    cash = cash + bet;
                    if (cash > maxcash) maxcash = cash;
                    printnotice("** YOU WIN EVEN MONEY **");
                    printfunds(cash, maxcash, bet);
                    break;
                case 2:
                    cash = cash + (2 * bet);
                    if (cash > maxcash) maxcash = cash;
                    printnotice("** YOU WIN 2 TO 1 **");
                    printfunds(cash, maxcash, bet);
                    break;
                case 3:
                    cash = cash + (3 * bet);
                    if (cash > maxcash) maxcash = cash;
                    printnotice("** YOU WIN 3 TO 1 **");
                    printfunds(cash, maxcash, bet);
                    break;
                default:
                    cash = cash - bet;
```

/* Listing continued on next page…*/

```
                    if (bet > cash) bet = cash;
                    printnotice("!! YOU LOSE !!");
                    printfunds(cash, maxcash, bet);
                    break;
            }
            break;

        case 'Q':
        case 'q':
            playing = FALSE;
            break;
        }

        refresh();
    }

    clearhelp();
    printoutro(cash);

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