Downhill

The chill mountain air whipping past you, the excitement of dodging trees while wearing two thin planks. Is there anything better than skiing?

How about skiing without leaving your desk? Just type in this short program and you're ready to race for the bottom of the slopes.

Use the arrow keys to move. Try to collect flags to increase your score. Each mountain is taller than the last, and stepper too so you go faster.

Downhill is written by James E. Ward.

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

#include<curses.h>
#include<time.h>
#include<stdlib.h>

#define YSIZE 24
#define XSIZE 25
#define PLROW 6
#define LIVES 5
#define CHANCE 16
#define FCHANCE 100
#define TURNS 100
#define LEFTMOST (COLS - XSIZE) / 2


chtype level[YSIZE][XSIZE];
int
lv=0,y=0,x=0,life=LIVES,px=XSIZE/2,score=0,chance=CHANCE,fchance=FCHANCE,turn=0;

float magicNumber= CLOCKS_PER_SEC;

chtype tree='&'|COLOR_PAIR(COLOR_GREEN);
chtype snow=' '|COLOR_PAIR(COLOR_WHITE);
chtype flag='P'|COLOR_PAIR(COLOR_MAGENTA)|A_BOLD;
chtype ltrail='/'|COLOR_PAIR(COLOR_WHITE)|A_BOLD;
chtype rtrail='\\'|COLOR_PAIR(COLOR_WHITE)|A_BOLD;
chtype dtrail='|'|COLOR_PAIR(COLOR_WHITE)|A_BOLD;

void empty();
void adv();
void init();
void die();
void draw();
void newlev();
void play();
void again();

void again()
{
    nodelay(stdscr,0);
    clear();
    mvprintw(0,0,"You made it to level %d with %d flags.\n"
      "Play again?(YN)", lv,score);
    refresh();
    switch (getch())
    {
    case 'Y':
    case'y':
        lv=0;
        life=LIVES;
        px=XSIZE/2;
        score=0;
        chance=CHANCE;
        fchance=FCHANCE;
        empty();
        clear();
        newlev();
        play();
        break;
    case 'N':
    case'n':
        exit(endwin());
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

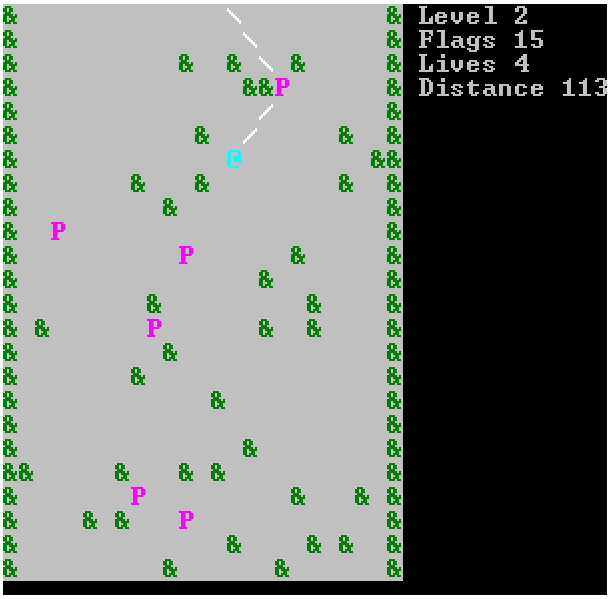
```
    default:
        again();
    }
}

void empty()
{
    for (y=0;y<YSIZE;y++)
    {
        for (x=0;x<XSIZE;x++)
        {
            level[y][x]=(x==0||x==XSIZE-1)?tree:snow;
        }
    }
}

void adv()
{
    for (y=0;y<YSIZE-1;y++)
    {
        for (x=0;x<XSIZE;x++)
        {
            level[y][x]=level[y+1][x];
        }
    }
    for (x=0;x<XSIZE;x++)
    {
        level[YSIZE-1][x]=(x==0||x==XSIZE-1||rand()%chance==0)
          ?tree:(rand()%fchance==0)?flag:snow;
    }
    if (level[PLROW][px]==flag)
    {
        score++;
        if (score%50==0)
        {
            life++;
        }
    }
    if (--turn<1)
    {
        newlev();
    }
    draw();
}

void init ()
{
    int c;
    srand (time(NULL));
    keypad(initscr(),1);
    raw ();
    nodelay(stdscr,0);
    noecho();
    curs_set(0);
    nonl();
    start_color();   // init color
    for (c=0; c<=8; c++)   // init color pairs
    {
        init_pair(c,c,COLOR_WHITE);
    }
}

void die()
```



```
Level 2
Flags 15
Lives 4
Distance 113
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
{
    beep();
    nodelay(stdscr,0);
    mvaddch(PLROW,LEFTMOST+px,'@'|COLOR_PAIR(COLOR_RED)|A_BOLD);
    px=XSIZE/2;
    turn+=YSIZE;
    empty();
    life--;
    getch();
    if (life<1)
    {
        again();
    }
    nodelay(stdscr,1);
}

void draw()
{

    for (y=0;y<YSIZE;y++)
    {
        for (x=0;x<XSIZE;x++)
        {
            mvaddch(y,LEFTMOST+x,level[y][x]);
        }
    }
    mvaddch(PLROW,LEFTMOST+px,'@'|COLOR_PAIR(COLOR_CYAN)|A_BOLD);
    mvprintw(0,LEFTMOST+XSIZE+1,"Level %d",lv);
    mvprintw(1,LEFTMOST+XSIZE+1,"Flags %d",score);
    mvprintw(2,LEFTMOST+XSIZE+1,"Lives %d",life);
    mvprintw(3,LEFTMOST+XSIZE+1,"Distance %d ",turn);
}

void newlev()
{
    nodelay(stdscr,0);
    empty();
    clear();
    lv++;
    px=XSIZE/2;
    if (chance>3)
    {
        chance--;
    }
    fchance--;
    turn=TURNS*lv;
    magicNumber = CLOCKS_PER_SEC /((float)lv+1);
    clear();
    nodelay(stdscr,1);
}

void play()
{
    nodelay(stdscr,1);
    int last=clock(),t=0;
    draw();
    while (1)
    {

        if (level[PLROW][px]==tree)
        {
            die();
        }
```

/* Listing continued on next page…*/

/* Listing continued from previous page */

```
        switch (getch())
        {
        case '4':
        case KEY_LEFT:
        case ',':
            last=clock();
            if (level[PLROW][px] == snow) level[PLROW][px]=ltrail;
            px--;
            adv();
            break;
        case'6':
        case KEY_RIGHT:
        case'.':
            last=clock();
            if (level[PLROW][px] == snow) level[PLROW][px]=rtrail;
            px++;
            adv();
            break;
        case'5':
        case' ':
        case KEY_DOWN:
            last=clock();
            if (level[PLROW][px] == snow) level[PLROW][px]=dtrail;
            adv();
            break;
        case'q':
        case'Q':
            again();
            break;
        default:
            break;
        }
        t=clock()-last;
        if (t>=magicNumber)
        {
            if (level[PLROW][px] == snow) level[PLROW][px]=dtrail;
            adv();
            last=clock();
        }
    }
}

int main(int argc,char* argv[])
{
    init();
    mvprintw(0,0,"Downhill ASCII\nBy James E. Ward June 2009");
    mvprintw(5,30,"   0    ");
    mvprintw(6,30,"  _|_   ");
    mvprintw(7,30," / | \\  ");
    mvprintw(8,30,"/ / \\ \\ ");
    mvprintw(9,30,"| \\  \\ |");
    mvprintw(10,30,"|  \\  \\|");
    mvprintw(14,0,"Use arrow keys to get the flags and dodge the trees");
    getch();
    newlev();
    play();
    exit(endwin());
}
```

/* Listing continued on next page…*/