A Journey to Hell

"A hero, right?", the handsome devil at the entrance asked. "Sign here, please", he boomed, presenting you with a sheet of paper.

One thing's for certain - you sure didn't expect Hell to have such good service. Like many before you, you were lured by stories of countless riches awaiting in the bowels of the hell realm. And the nice lawyer at the gates assures you that you are free to take as much as you can carry - as long as you are able to walk out with it.

Reach the seventh order of Hell, collecting as much gold as possible on the way. Each level is full of gold and chests which contain numerous valuables. Of course, Hell also has its guardians. Demons roam freely, and the deeper you go, the more powerful they become. Fortunately, they are quite stupid - they are easily distracted and are often found staring at a shiny object or contemplating a set of stairs. Once they get close to you, though, they make up with brawn what they lack in brains. Be careful, they are also very fast - unlike most roguelikes, the demons move BEFORE the PC.

Hitting a demon will decrease its power, until it is dead. A vanquished demon leaves behind souls of all the heroes it devoured. These souls can enhance your own spirit and increase your resilience.

/* Name.c linsting begins: */

```
/* This version has been modified from the original commented version
 * by Jakob Wadellishimyer for Cymon's Games
 * to fix the random seeding
 * and unify the color initilazition between versions of curses.
 */
#include <curses.h>
#include <stdlib.h>
#include <time.h>

/* 'while' short form */
#define W(c)while(c)
/* 'for' short form */
#define F(v,h)for(v=0;v<h;v++)
/* launch instruction 'i' for every tile in the dungeon */
#define D(i)F(y,S)F(x,T){i;}
/* short form for dungeon tile at position x,y */
#define G d[x][y]
/* short form for dungeon tile at position x+f,y+g - useful for movement/
generation */
#define M d[x+f][y+g]
/* sign of number 'n' */
#define Z(n)n<0?-1:(n>0?1:0)
/* random number */
#define r rand()

/*
  Variables:
  x,y - always indicate a position in the dungeon, used mostly through G
  c   - helper variable, stores pressed character, used for color pairs initiali-
zation
  d[][] - dungeon map
  L   - current dungeon level
  l,p - HP current/maximum
  v   - gold collected
  f,g - secondary dungeon position, used in generation/movement
  h,i - room dimensions, used in generation
  b,z - temporary variables

  Constants:
  S,T - size of dungeon, SxT
  e   - a constant meaning '.'
  u   - a constant used for monster handling, 'a'-1
  k[] - a table used for coloring output, putting stuff in the dungeon, and doing
chests
*/
int x,y,S=24,T=60,c,d[90][90],L=1,b,
  f,g,z,h,i,l=9,p=20,e=46,v,u=96;
wchar_t k[]=L"$`*$.@?$?>";

/* find a random empty position and insert t in it */
int E(int t){W(d[x=r%T][y=r%S]-e);G=t;}

int N(){
  /* fill with walls */
  D(G=35)
  /* add rooms */
  x=y=9;
  F(b,S)
  {
    f=r%58+1;g=r%22+1;
    W(f^x||g^y){
```

/* Listing continued on next page… */

/* Listing continued from previous page */

```
      G=e,
      r&1?(x+=Z(f-x)):(y+=Z(g-y));
    }
  }
  /* put stairs, PC, blood gem */
  F(b,10)E(k[b]);
  /* put monsters */
  F(b,L*2)E(98+r%L);
}
/* move the actor at (x,y) by [f,g] */
int m(int f, int g){
  /* is player? */
  G-64?
    /* not a player, so a monster */
    /* monster moved into player? */
    M-64||(l-=G-u)>0||(M=e)
  :
  (
    /* player code */
    /* stairs */
    M^62||L++&&(L^7?N():(G=e)),
    /* talismans */
    M^u||(p+=L,M=e),
    /* blood gems */
    M^42||(l=p,M=e),
    /* gold */
    M^36||(v+=x%9*L+L,M=e),
    /* chests */
    M^63||(M=r%6?k[r&1]:97+L),
    /* attacking monsters */
    M>u&&--M
  )
  ;

  /* actual movement */
  M-e?M-35||(g&&m(f,0),f&&m(0,g)):(M=G|256,G=e);
}

/* main */
int main()
{
  /* curses/srand initialization */
  srand(time(NULL));
  initscr();raw();start_color();
  /* set up colors */
  F(c,8)
#ifdef PDCURSES
  init_pair(c,(c&2)|((c&4) / 4)|((c&1) * 4),0);
#else
  init_pair(c,c,0);
#endif

  /* generate a level */
  N();

  /* play until Q is pressed */
  W(c-81)
  {
    /* find player, store position in f,g */
    D(G-64||(f=x,g=y))

    /* draw screen */
    clear();
```

/* Listing continued on next page… */

1-9 on the numeric keypad is used for movement (NumLock HAS TO be on). 5 can be used to wait. Q (that's SHIFT+q) quits the game.

Items are picked up automatically, and other features are used by walking into them. After the game ends (due to death or winning), you still have to press SHIFT+Q to quit.

Key:

@ This is the player character.

a-z These are demons, with 'a' being the weakest, and higher letters being progressively more powerful.

` This is a soul. Collect those to increase your maximum HP.

* These are power gems. Each level has one, and taking it will restore your HP to maximum (one use only).

$ This is gold. Collect as much as you can to increase your final score.

? These are magical chests, walk into them to open them. They usually contain gold or souls, but once in a while they can also have a nasty surprise inside.

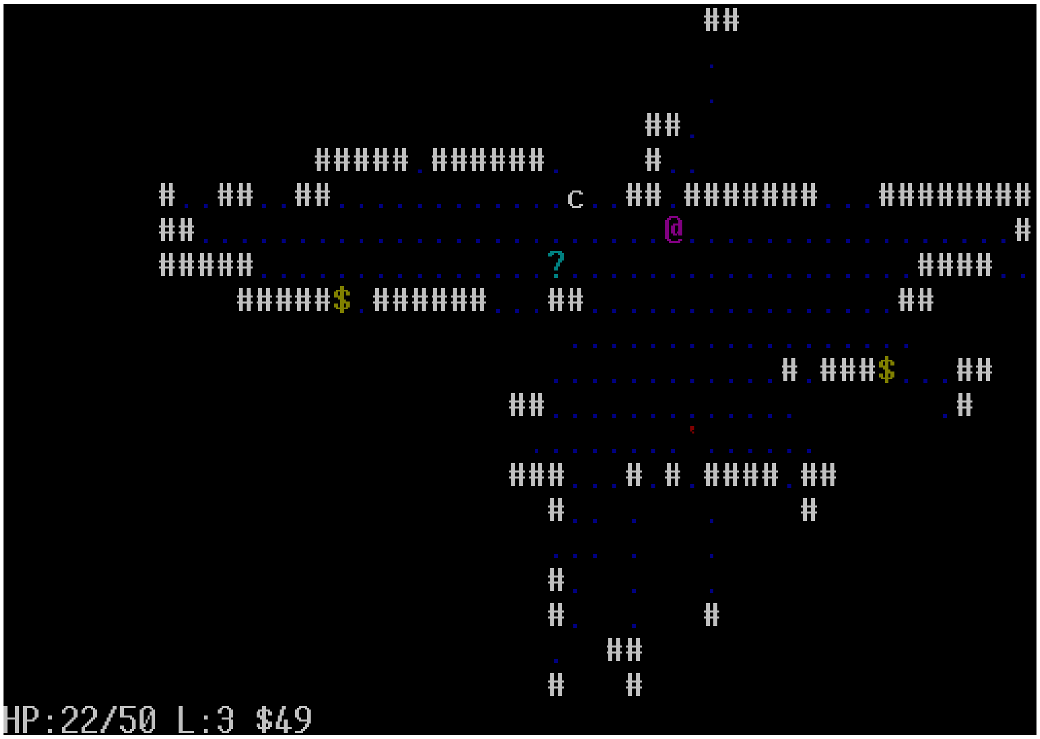
TIPS:

- always remember that the demons move BEFORE your character
- The souls in Hell are more powerful on deeper levels - they will grant a greater HP increase- maximize the benefits of power gems by taking them as late as possible without dying

- Never fight more than one monster at a time

A Journey to Hell was written by Jakub Wasilewski to use exactly 1k of code (1024 characters). This version has been modified from the original commented version to fix the random seeding and unify the color initialization between versions of curses.

```
/* Listing continued from previous page */
    D(
      /* determine tile colour */
      b=1;W(G^k[b]&&b<7)b++;
      /* check for visibility */
      h=x;i=y;c=0;W(c++<u)(d[f+(z=(h-f)*c*.01)][g+(z=(i-g)*c*.01)]-35)||(b=0);
      /* draw, if visible at all */
      b&&mvaddch(y,x,G|COLOR_PAIR(b))
    )

    /* status bar */
    mvprintw(S,0,"HP:%d/%d L:%d $%d %s",l,p,L,v,L^7?"":"WIN");

    /* get input */
    c=getch();

    /* move monsters */
    D(G>u&&G<123&&m(Z(f-x),Z(g-y)))

    /* move player, if needed */
    D(c<49||c>57||G-64||m((c-1)%3-1,1-(c-49)/3))

    /* clear move flags for player and monsters */
    D(G&=255)
  }

  /* clean up */
  endwin();
}
```



```
HP:22/50 L:3 $49
```