

Get Out!

You have stolen the famous Rogue Diamond from the national museum, and hidden it cunningly about your person. But with the roof entrance you arrived through now out of reach, you have to escape through the museum itself. There are ten floors with stairs (>) on each level, but you have to retrieve the security keycard (!) to unseal each exit.

The guards may pose a problem, but as the world's greatest thief that shouldn't be a challenge.

Move your character (@) using the number pad. You can move in any of 8 directions or press the '5' key to hold still for a moment. The cop (&) moves predictably and can be trapped but will attempt to go around any obstacles in his path. To disable the cops ability to dodge obstacles you can press 'X' on the main menu to change the cops behavior.

Get out is written by James E. Ward.

```
/* class.h listing begins: */
class map
{
public:
    //20 changed to 25 for test
    char show[25][40], card;
    map();
    int exity,exitx,cardx,cardy;
};
```

```
class people
{
public:
    char sym;
    int x,y;
};
```

```
/* map.cpp listing begins: */
```

```
#include <iostream>
#include <stdlib.h>
#include "class.h"
#include <time.h>
```

```
using namespace std;
```

```
map::map()
{
    int z;
```

```
    exity=rand()%18+1;
    exitx=rand()%38+1;
    cardy=rand()%18+1;
    cardx=rand()%38+1;
```

```
    for (int yy=0;yy<20;yy++)
    {
```

```
        for (int xx=0;xx<40;xx++)
        {
```

```
            show[yy][xx] = (xx==0 || xx==39 || yy==0 || yy==19) ? '*' : '.';
            z=rand() % 6;
            if (z==4 && (xx!=1 || yy!=1))
                show[yy][xx] = '*';
        }
    }
```

```
    if (show[exity][exitx] == '*')
```

```
        show[exity][exitx]=' ';
```

```
    if (show[cardy][cardx] == '*')
```

```
        show[cardy][cardx]=' ';
```

```
    card='!';
```



```
/* main.cpp listing begins: */
```

```
#include <string>
#include <cstdlib>
#include <curses.h>
#include <time.h>
#include <math.h>
#include "class.h"
```

```
#define MV(K,Y,X) if(K==c && '*'!=lv.show[you->Y][you->X])
```

```
using namespace std;
```

/* Listing continued on next page... */

```

/* Listing continued from previous page */
//declarations

bool easy=true,card=false;
int level=0,turn=0;
void genpolice(people* pig,map* lv,people* you);
void init ();
void play();
void win();
void lose();
void mpig(people* pig,people* player,map* lv);
bool input(char, map,people*);
void replay();
void help();
void drscrn(map lv, people player, people police);
void again();

void again()
{
    char n;
    for (;;)
    {
        mvprintw(24,0,"Play again? (y/n)");

        if ('Y'==(n=getch()) || n=='y')
        {
            clear();
            turn=0;
            level=0;
            play();
        }
        else if (n=='n' || n=='N' || n=='q')
        {
            exit(endwin());
        }
    }
}

void drscrn(map lv, people player, people police)
{
    clear();
    //draw map
    for (int yy=0;yy<20;yy++)
        for (int xx=0;xx<40;xx++)
            mvaddch(yy,xx,lv.show[yy][xx]);
    //draw everything else on the screen
    mvaddch(lv.exity,lv.exitx, '>'|COLOR_PAIR(COLOR_GREEN));

    mvaddch(lv.cardy, lv.cardx,lv.card|COLOR_PAIR(COLOR_YELLOW));

    mvaddch(player.y,player.x,player.sym|COLOR_PAIR(COLOR_CYAN));
    mvaddch(police.y,police.x,police.sym|COLOR_PAIR(COLOR_CYAN));
    if (card)
    {
        mvaddch(21,0,'!'|COLOR_PAIR(COLOR_YELLOW));
    }
    else
    {
        mvaddch(21,0,' ');
    }
    mvprintw(22,0,"Level: %d", level+1);
    mvprintw(23,0,"Turn: %d", turn);
}

```

/* Listing continued on next page... */

```

/* Listing continued from previous page */

//help screen
void help()
{
    clear();
    mvprintw(1,20,"Get Out Help Page");
    mvprintw(3,0,"Objective:");
    mvprintw(4,0,"Escape all 10 levels of the museum, first getting the key then");
    mvprintw(5,0,"reaching the stairs while evading the police.");
    mvprintw(7,0,"Symbols:");
    mvaddch(8,0,'@'|COLOR_PAIR(COLOR_CYAN));
    mvprintw(8,3,"= Player");
    mvaddch(9,0,'&'|COLOR_PAIR(COLOR_CYAN));
    mvprintw(9,3,"= The big bad cop");
    mvaddch(10,0,'!'|COLOR_PAIR(COLOR_YELLOW));
    mvprintw(10,3,"= Keycard");
    mvaddch(11,0,'>'|COLOR_PAIR(COLOR_GREEN));
    mvprintw(11,3,"= Stairs to next level");
    mvprintw(12,0,". = floor");
    mvprintw(13,0,"* = Wall");
    mvprintw(15,0,"Controls:");
    mvprintw(16,0,"Movement Keys:");
    mvprintw(17,0,"789 yku");
    mvprintw(18,0,"456 or h.l");
    mvprintw(19,0,"123 bjn");
    mvprintw(21,0,"'q' = Quit game");
    mvprintw(22,0,"'x' or 'X' at intro screen for hard mode");
    mvprintw(23,0,"'? = Help screen");

    getch();
    clear();
}

void init ()
{
    srand (time(NULL));

    keypad(initscr(),1);//initializes the keypad and starts curses mode
    start_color(); // init color
    for (int c=0; c<=8; c++) // init color pairs
    {
        init_pair(c,c,0);
    }
#ifdef PDCURSES
    PDC_set_title ("Get Out by James E. Ward on Cymon's Games");
#endif

    raw ();
    noecho();
    curs_set(0);
    nonl();
}

// player input
bool input(char c,map lv, people* you)
{
    MV('7',y-1,x-1)
    {
        you->y--;
        you->x--;
        return true;
    }
    MV('y',y-1,x-1)
    {

```

/* Listing continued on next page... */

/* Listing continued from previous page */

```

    you->y--;
    you->x--;
    return true;
}
MV('8',y-1,x)
{
    you->y--;
    return true;
}
MV('k',y-1,x)
{
    you->y--;
    return true;
}
MV('9',y-1,x+1)
{
    you->y--;
    you->x++;
    return true;
}
MV('u',y-1,x+1)
{
    you->y--;
    you->x++;
    return true;
}
MV('4',y,x-1)
{
    you->x--;
    return true;
}
MV('h',y,x-1)
{
    you->x--;
    return true;
}
MV('6',y,x+1)
{
    you->x++;
    return true;
}
MV('l',y,x+1)
{
    you->x++;
    return true;
}
MV('1',y+1,x-1)
{
    you->y++;
    you->x--;
    return true;
}
MV('b',y+1,x-1)
{
    you->y++;
    you->x--;
    return true;
}
MV('2',y+1,x)
{
    you->y++;
    return true;
}

```



/* Listing continued on next page... */

/* Listing continued from previous page */

```
MV('j',y+1,x)
{
    you->y++;
    return true;
}
MV('3',y+1,x+1)
{
    you->y++;
    you->x++;
    return true;
}
MV('n',y+1,x+1)
{
    you->y++;
    you->x++;
    return true;
}
MV('5',y,x)
{
    you->y;
    you->x;
    return true;
}
MV('.',y,x)
{
    you->y;
    you->x;
    return true;
}
MV('?',y,x)
{
    help();
    return false;
}
else
{
    return false;
}
}
```

// heres the meat

```
void play()
{
    //setup
    map maps[10];

    //to prevent trapping
    for (int i=0;i<9;i++)
    {
        maps[i+1].show[maps[i].exity][maps[i].exitx] = '.';
        if (maps[i].exitx !=1)
        {
            maps[i+1].show[maps[i].exity][maps[i].exitx-1]='.';
        }
        if (maps[i].exitx !=38)
        {
            maps[i+1].show[maps[i].exity][maps[i].exitx+1]='.';
        }
        if (maps[i].exity !=1)
        {
            maps[i+1].show[maps[i].exity-1][maps[i].exitx]='.';
        }
    }
}
```

/* Listing continued on next page... */

/* Listing continued from previous page */

```
    if (maps[i].exity !=18)
    {
        maps[i+1].show[maps[i].exity+1][maps[i].exitx]='.';
    }
}
maps[0].show[1][2]='.';
maps[0].show[2][1]='.';

//continuing setup
people player;
player.sym='@';
player.x=1;
player.y=1;
card = false;
people police;
police.sym='&';
genpolice(&police, &maps[level],&player);
int c;
drscrn(maps[level],player,police);

//main game loop
do
{
    // get input from player & do all of this stuff
    if (input(c,maps[level],&player))
    {
        //go to next level
        if (card && player.y == maps[level].exity && player.x
            == maps[level].exitx)
        {
            if (level==9)
            {
                win();
            }
            level++;
            card=false;
            genpolice(&police,&maps[level],&player);
            for (int yy=0;yy<20;yy++)
                for (int xx=0;xx<40;xx++)
                    mvaddch(yy,xx,maps[level].show[yy][xx]);
        }
        /* move the police. If you have the keycard, you can escape him
        by getting to the stairs that is why this is after the next
        level code*/
        mpig(&police,&player,&maps[level]);
        //and increase the turn counter
        turn++;
    }

    //pick up the key card
    if (player.y == maps[level].cardy && player.x == maps[level].cardx)
    {
        maps[level].card='.';
        card=true;
    }
    //draw the screen
    drscrn(maps[level],player,police);

    //check for loss (better here)
    if (police.y==player.y&&police.x==player.x)
    {
        lose();
    }
}
```

/* Listing continued on next page... */

/* Listing continued from previous page */

```
    }
    while ('q'!=(c=getch()));
    again();
}

void genpolice(people* pig,map* lv,people* you)
{
    //generates the location of the police
    double xdist,ydist,dist;
    do
    {
        pig->x=rand() % 38 + 1;
        pig->y=rand() % 18 + 1;
        ydist = pig->y - you->y;
        xdist = pig->x - you->x;
        dist = sqrt((ydist*ydist)+(xdist*xdist));
    }
    while (lv->show[pig->y][pig->x] != '.' || dist <= 8);
}

void win()
{
    getch();
    clear();
    endwin();
    //Win text by Darren Grey
    mvprintw(2,0,"Congratulations! You have escaped the museum in one piece,");
    mvprintw(3,0,"and after sanitising the diamond thoroughly it fetches a "
        "whopping");
    mvprintw(4,0,"$%d on the black market! Now for a life of luxury in the "
        "Bahamas...", 1000000000/turn);
    getch();
    again();
}

void lose()
{
    getch();
    clear();

    //Darren Grey wrote this loss text
    mvprintw(2,0,"You've been nicked! After an invasive body search the "
        "diamond is");
    mvprintw(3,0,"retrieved, and you are sentenced to %d years in the slammer."
        , level+11);
    mvprintw(4,0,"Instead of luxury you must now face a life of poor food and "
        "wary showers...");
    again();
}

void mpig(people* pig,people* you, map* lv)
{
    bool pl = (pig->x-you->x>0);//if you are to the left of the pig
    bool pr = (pig->x-you->x<0);//if you are to the right of the pig
    bool pu = (pig->y-you->y>0);//if you are above the pig
    bool pd = (pig->y-you->y<0);//if you are below the pig
    bool pul = (pu && pl);//if you are above and to the left of the pig
    bool pur = (pu && pr);//if you are above and to the right of the pig
    bool pdl = (pd && pl);//if you are below and to the left of the pig
    bool pdr = (pd && pr);//if you are below and to the right of the pig
    for (;;)
```

/* Listing continued on next page... */

/* Listing continued from previous page */

```
{
  if (pdl && lv->show[pig->y+1][pig->x-1]!='*')
  {
    pig->x--;
    pig->y++;
    break;
  }
  else if (pul && lv->show[pig->y-1][pig->x-1]!='*')
  {
    pig->x--;
    pig->y--;
    break;
  }
  else if (pur && lv->show[pig->y-1][pig->x+1]!='*')
  {
    pig->x++;
    pig->y--;
    break;
  }
  else if (pdr && lv->show[pig->y+1][pig->x+1]!='*')
  {
    pig->x++;
    pig->y++;
    break;
  }
  else if (pl && lv->show[pig->y][pig->x-1]!='*')
  {
    pig->x--;
    break;
  }
  else if (pr && lv->show[pig->y][pig->x+1]!='*')
  {
    pig->x++;
    break;
  }
  else if (pu && lv->show[pig->y-1][pig->x]!='*')
  {
    pig->y--;
    break;
  }
  else if (pd && lv->show[pig->y+1][pig->x]!='*')
  {
    pig->y++;
    break;
  }
  else if (easy)
  {
    //Expanded AI is actually easier to beat
    if (pd && lv->show[pig->y+1][pig->x+1]!='*')
    {
      pig->y++;
      pig->x++;
      break;
    }
    else if (pd && lv->show[pig->y+1][pig->x-1]!='*')
    {
      pig->y++;
      pig->x--;
      break;
    }
    else if (pr && lv->show[pig->y+1][pig->x+1]!='*')
    {
      pig->y++;
    }
  }
}
```



/* Listing continued on next page... */

/* Listing continued from previous page */

```
        pig->x++;
        break;
    }
    else if (pr && lv->show[pig->y-1][pig->x+1]!='*')
    {
        pig->y--;
        pig->x++;
        break;
    }
    else if (pu && lv->show[pig->y-1][pig->x+1]!='*')
    {
        pig->y--;
        pig->x++;
        break;
    }
    else if (pu && lv->show[pig->y-1][pig->x-1]!='*')
    {
        pig->y--;
        pig->x--;
        break;
    }
    else if (pl && lv->show[pig->y+1][pig->x-1]!='*')
    {
        pig->y++;
        pig->x--;
        break;
    }
    else if (pl && lv->show[pig->y-1][pig->x-1]!='*')
    {
        pig->y--;
        pig->x--;
        break;
    }
    else if (pdl && lv->show[pig->y-1][pig->x]!='*')
    {
        pig->y--;
        break;
    }
    else if (pdl && lv->show[pig->y][pig->x+1]!='*')
    {
        pig->x++;
        break;
    }
    else if (pdr && lv->show[pig->y--][pig->x]!='*')
    {
        pig->y--;
        break;
    }
    else if (pdr && lv->show[pig->y][pig->x-1]!='*')
    {
        pig->x--;
        break;
    }
    else if (pul && lv->show[pig->y+1][pig->x]!='*')
    {
        pig->y++;
        break;
    }
    else if (pul && lv->show[pig->y][pig->x+1]!='*')
    {
        pig->x++;
        break;
    }
}
```

/* Listing continued on next page... */

/* Listing continued from previous page */

```
        else if (pur && lv->show[pig->y+1][pig->x]!='')
        {
            pig->y++;
            break;
        }
        else if (pur && lv->show[pig->y][pig->x-1]!='')
        {
            pig->x--;
            break;
        }
        else break;
    }
    else break;
}

int main()
{
    //Thanks to Darren Grey for this introduction
    init();

    mvprintw(1,20,"Welcome to GET OUT!!");
    mvprintw(3,0,"You have stolen the famous Rogue Diamond from the national "
        "museum,");
    mvprintw(4,0,"and hidden it cunningly about your person. But with the roof "
        "entrance");
    mvprintw(5,0,"you arrived through now out of reach, you have to escape "
        "through the");
    mvprintw(6,0,"museum itself. There are ten floors with stairs (>) on each "
        "level,");
    mvprintw(7,0,"but you have to retrieve the security keycard (!) to unseal "
        "each exit.");
    mvprintw(8,0,"The guards may pose a problem, but as the world's greatest "
        "thief they,");
    mvprintw(9,0,"shouldnt be a challenge.");

    char dif;

    if ('x'==(dif=getch())||dif=='X')
    {
        easy=false;
    }
    play();
    exit(endwin());
}
```