四時讀書樂 (春)

山光照檻水繞廊，舞雩歸詠春風香。
好鳥枝頭亦朋友，落花水面皆文章。
蹉跎莫遣韶光老，人生唯有讀書好。
讀書之樂樂何如，綠滿窗前草不除。

~ 元·翁森

四時讀書樂 (夏)

新竹壓檐桑四圍，小齋幽敞明朱暉。
晝長吟罷蟬鳴樹，夜深燼落螢入帷。
北窗高臥羲皇侶，只因素稔讀書趣。
讀書之樂樂無窮，瑤琴一曲來薰風。

~ 元·翁森

四時讀書樂 (秋)

昨夜庭前葉有聲，籬豆花開蟋蟀鳴。
不覺商意滿林薄，蕭然萬籟涵虛清。
近床賴有短檠在，趁此讀書功更倍。
讀書之樂樂陶陶，起弄明月霜天高。

~ 元·翁森

四時讀書樂 (冬)

木落水盡千崖枯，迥然吾亦見真吾。
坐對韋編燈動壁，高歌夜半雪壓廬。
地爐茶鼎烹活火，四壁圖書中有我。
讀書之樂何處尋？數點梅花天地心。

~ 元·翁森

Some Useful Escape Sequence

- \n - newline (a.k.a. line feed, LF)
- \t - horizontal tab
- \v - vertical tab
- \r - carriage return (CR)
- \b - backspace
- \f - clear screen (a.k.a. form feed, FF)

\n \v \r \b

```
#include <iostream>
#include <unistd.h>

int main()
{
    const char* words[] = { "May", "the", "force", "be", "with", "you" };
    const char c = '\n';        // \n \v \r \b
    for (size_t i = 0; i < sizeof(words)/sizeof(words[0]); ++i) {
        std::cout << words[i] << c << std::flush;
        usleep(1000000);        // microsecond
    }

    std::cout << std::endl;
    return 0;
}
```

Similar in the Old C

```
#include <stdio.h>
#include <unistd.h>

int main()
{
    const char* words[] = { "May", "the", "force", "be", "with", "you" };
    const char c = '\v';        // \n \v \r \b
    for (size_t i = 0; i < sizeof(words)/sizeof(words[0]); ++i) {
        printf("%s%c", words[i], c);
        // fflush(stdout);
        usleep(1000000);        // microsecond
    }

    printf("\n");
    return 0;
}
```

~solomon/CPP/showtime.exe

```
#include <iostream>
#include <ctime>
#include <unistd.h>

int main() {
    time_t t;
    struct tm* now;
    char buf[9];
    while (true) {
        time(&t);
        now = localtime(&t);
        strftime(buf, sizeof(buf), "%T", now);
        std::cout << '\r' << buf << std::flush;
        sleep(1);
    }
    return 0;
}
```

Exercise: Moving Time

- ~solomon/CPP/movetime.exe

NCURSES Library

- Functions for Screen Handling
- Visual C++ 2019 (P.10-P.46)
  - g++ on Linux (P.47)

Ncurses Library on Unix

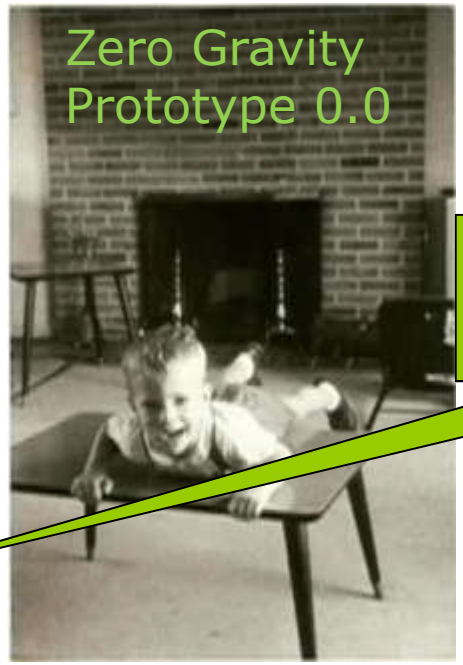
- *New* Cursor Control library
- `man ncurses`
- Linux Journal - Getting Started with ncurses
  - https://www.linuxjournal.com/content/getting-started-ncurses

Moving a Star Towards the Left

- Suppose we want to have a program, which display a star on the screen.
- With a fixed interval, the star will move towards the left, step by step.
  - ncurses-1.exe

ncurse-1.cpp

```
#include <iostream>
#include <unistd.h>
using std::cout;
using std::endl;

int main()
{
    int i;
    int j;
    for (i=20; i>=0; i--)
    {
        for (j=0; j<i; j++)
            cout << ' ';
        cout << '*' ;
        for (j=0; j<23; j++)
            cout << endl;
        usleep(300000);  // Pause for 0.3sec
    }
    return 0;
}
```



What if the window is shrunk to fewer rows?

If the moving pattern is complex …

- ~solomon/CPP/Ncurses/ncurses-2.exe
  - (optional) Inspect ncurses-2.cpp
- Your program will become very complex, if you try to implement as the way in the previous program.
- It would be great if we can move the cursor to wherever we want, by specifying the coordinate.

Definition of **curses** on Wikipedia

- **curses** is
  a terminal control library for Unix-like systems, enabling the construction of text user interface (TUI) applications.
- The name is a pun on the term "cursor optimization". It is a library of functions that manage an application's display on character-cell terminals (e.g., VT100)

Basic Functions

- move(y,x)
  - Move cursor to (y,x) in screen
  - The coordinates are zero-based.
- addch(ch)
  - Add a character to screen.
  - X coordinate will be incremented
- addstr(str)
  - Add a string to screen by calling addch()
- printw(fmt, arg1, arg2, …)
  - Formatted print to screen by calling addstr()
- refresh()
  - Update screen

Initialize and Terminate Curses

- initscr()
  - Initialize curses
- endwin()
  - End curses.
  - This function should be called when your program is finished.
  - It will release the space allocated to screen handling in your program.

- Remember to `#include <curses.h>` at the beginning of your program.
  - In many cases, you don't need `<iostream>`.

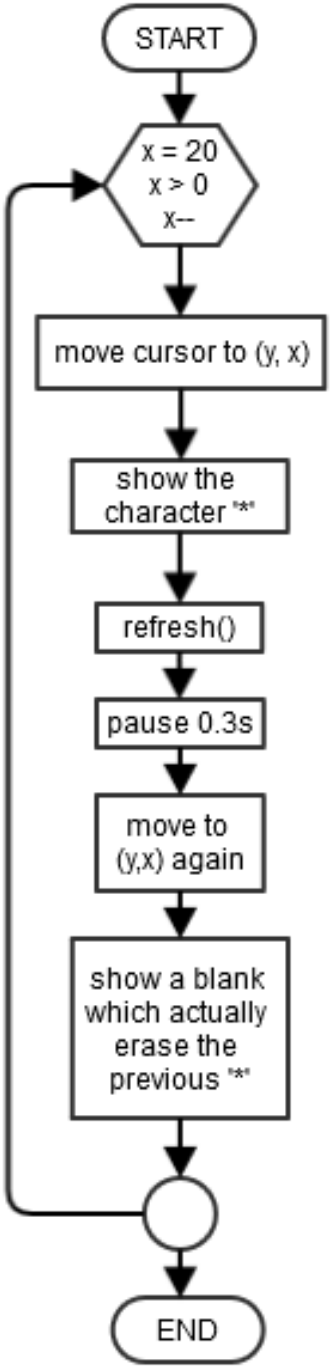
ncurses-3.cpp

```
#include <curses.h>

int main()
{
    initscr();
    addch('=');
    addstr("Hello\n");
    printw("%d\t%d\n", LINES, COLS);
    move(3, 3); addch('A');
    move(2, 2); addch('B');
    refresh();
    // getch();
    endwin();
}
```

- Compile your program and link with the curses library
  - g++ ncurses-3.cpp -lcurses -o ncurses-3.exe

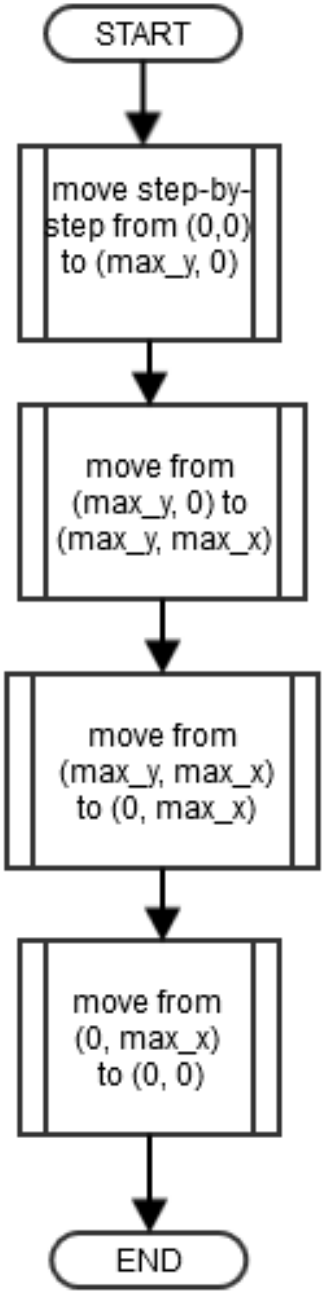
Hands-On: A Moving Star

- Use the "curses" library to write a program, which shows a star moving towards the left



Homework: Move Around

- Since you learned how to move a star, extend the exercise to move the star around a rectangle.
  - You may need 4 loops to do that.
  - Run "ncurses-2.exe" again to show students what they must achieve.



Getting Input from the Terminal

- getch()
  - Get a character from the terminal
- getstr(str)
  - Get a string from the terminal
- scanw(fmt, arg1, arg2, …)
  - Formatted input from the terminal like scanf().

ncurses-4.cpp: getstr() vs. scanw()

```
#include <curses.h>

int main()
{
   char text[10];
   int i, j, c;
   initscr();
   getstr(text);                  // input the string "1,2"
   addstr(text); addch('\n');

   scanw("%d,%d", &i, &j);        // input the string "1,2" again
   printw("%d\t%d\n", i, j);

   c = getch();
   endwin();
   return 0;
}
```

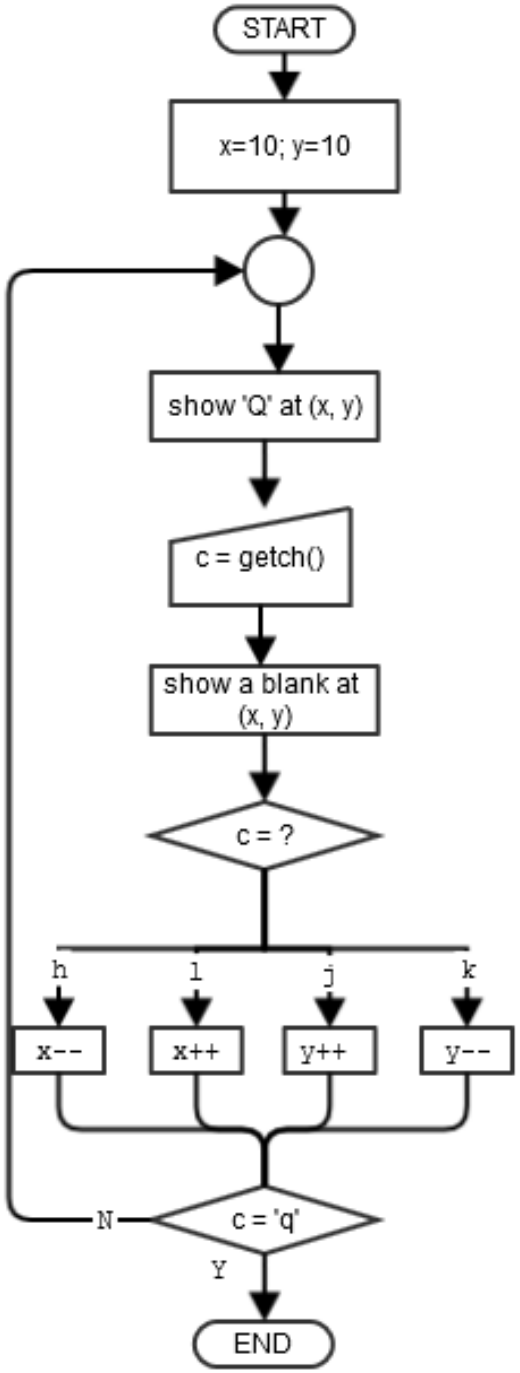
ncurses-5.cpp

```
#include <curses.h>

int main()
{
    int i;
    char c;
    initscr();
    for (i=0; i<5; i++) {
        c = getch();
        switch (c) {
            case 'h':
                addstr("Left");
                break;
            case 'l':
                addstr("Right");
                break;
            }
        refresh();
        }
    endwin();
    return 0;
}
```

ncurses-5a.cpp

```
#include <curses.h>

int main()
{
    int i;
    char c;
    initscr();
    noecho();
    for (i=0; i<5; i++) {
        c = getch();
        switch (c) {
            case 'h':
                addstr("Left");
                break;
            case 'l':
                addstr("Right");
                break;
            }
        refresh();
        }
    endwin();
    return 0;
}
```

h,j,k,l for Left,Down,Up,Right



ncurses-6.cpp

```
// ncurses-6.cpp
#include <curses.h>

int main()
{
    int y=10, x=10;
    char c;
    initscr();
    noecho();
    do {
          move(y, x); addch('Q');
          refresh();
          c = getch();
          move(y, x); addch(' ');
          switch (c)
          {
          case 'h':
                   x--;
                   break;
          case 'l':
                   x++;
                   break;
          case 'j':
                   y++;
                   break;
          case 'k':
                   y--;
                   break;
          }
    } while (c != 'q');

    endwin();
    return 0;
}
```

curs_set()

- `curs_set()` alters the appearance of the text cursor.
- `int curs_set(int visibility);`
  - A value of 0 for visibility makes the cursor disappear;
  - A value of 1 makes the cursor appear "normal" (usually an underline)
  - 2 makes the cursor "highly visible" (usually a block).

ncurses-6a.cpp

```
// ncurses-6a.cpp
#include <curses.h>

int main()
{
    int y=10, x=10;
    char c;
    initscr();
    noecho();
    curs_set(0);   // no cursor
    do {
          move(y, x); addch('Q');
          c = getch();
          move(y, x); addch(' ');
          switch (c)
          {
          case 'h':
                   x--;
                   break;
          case 'l':
                   x++;
                   break;
          case 'j':
                   y++;
                   break;
          case 'k':
                   y--;
                   break;
          }
    } while (c != 'q');

    endwin();
    return 0;
}
```

Exercise: Show Time

- Depending on the size of the PuTTY window, when your program starts, it should show the current time at the center of the screen.
- Press h,j,k,l will move the time to be shown in a new position which is at the left, down, up, right position of the previous position.

Game of Tanks

- With these skills, you can actually design a game similar to World of Tanks (although less fancy).

- ncurses-7.exe
  - h,j,k,l for Left,Backward,Foreward,Right
  - Spacebar to cannon.

Keypad Keys

- Call `keypad()` to enable the handling of keypad keys and arrow keys.
  - `int keypad(WINDOW *win, bool bf);`
    - `keypad(stdscr, TRUE);`
- `getch()` returns an integer corresponding to the key pressed.
  - If it is a normal character, the integer value will be equivalent to the ASCII code of the character.
  - Otherwise it returns a number which can be matched with the constants defined in `curses.h`.
    - For example if the user presses KEY_DOWN, the integer returned is 258.

Keypad Keys (cont.)

- With keypad() enabled, you can check the returned value of getch() with the constants defined in curses.h
  - KEY_DOWN, KEY_UP, KEY_LEFT, KEY_RIGHT
  - KEY_IC (Insert character)
  - KEY_DC (Delete character)
  - KEY_NPAGE (next page, i.e., Page Down)
  - KEY_PPAGE (previous page, i.e., Page Up)
  - KEY_F6 ~ KEY_F12

Key Definitions

```
#define KEY_DOWN        0402  /* down-arrow key */
#define KEY_UP          0403  /* up-arrow key */
#define KEY_LEFT        0404  /* left-arrow key */
#define KEY_RIGHT       0405  /* right-arrow key */

#define KEY_IC          0513  /* insert char or enter ins
mode (Insert) */
#define KEY_DC          0512  /* delete character (Delete)
*/
#define KEY_PPAGE       0523  /* previous page (PageUp) */
#define KEY_NPAGE       0522  /* next page (PageDown) */
```

You may check /usr/include/curses.h to see more definitions.

ncurses-8.cpp

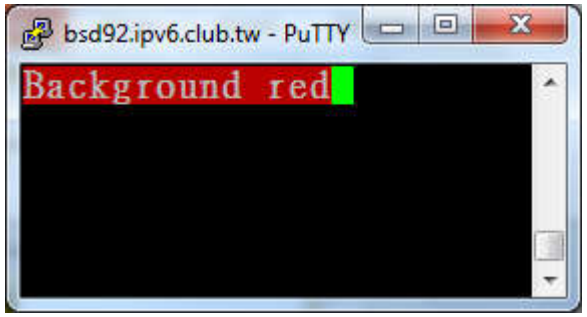
```
#include <curses.h>

int main()
{
        int c;
        initscr();
        noecho();
        keypad(stdscr, TRUE);
        do {
                move(9, 10);
                printw("Press a key ('q' to quit):");
                c = getch();
                move(10, 10);
                printw("%04o %c\n", c, c);
                refresh();
        } while (c != 'q');

        endwin();
        return 0;
}
```

Colors

- To start using color, you should first call the function `start_color()`.
  - To find out whether a terminal has color capabilities or not, you can use `has_colors()` function, which returns FALSE if the terminal does not support color.
- Colors are always used in pairs.
  - A color-pair consists of a foreground color and a background color.
  - Initializes a color-pair with the routine `init_pair()`. After it has been initialized, `COLOR_PAIR(n)` is used to represent the color attribute.

ncurses-9.cpp

```
#include <curses.h>

int main()
{
        initscr();
        start_color();

        init_pair( 1, COLOR_WHITE, COLOR_RED );
        attron( COLOR_PAIR(1) );
        printw("Background red");
        attroff( COLOR_PAIR(1) );

        refresh();
        getch();
        endwin();
        return 0;
}
```



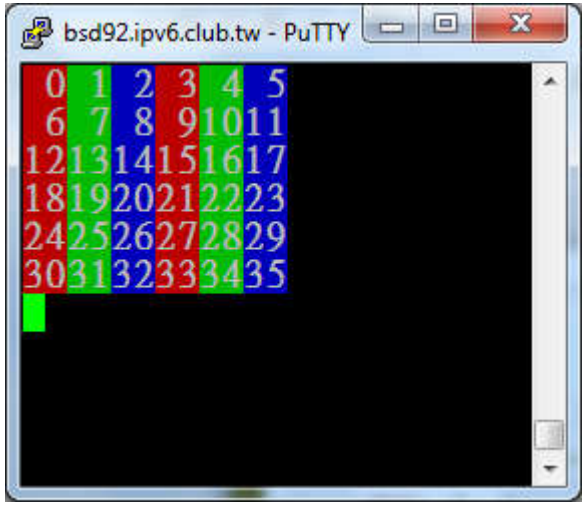
Pre-defined Colors on Unix

- COLOR_BLACK = 0
- COLOR_RED = 1
- COLOR_GREEN = 2
- COLOR_YELLOW = 3
- COLOR_BLUE = 4
- COLOR_MAGENTA = 5
- COLOR_CYAN = 6
- COLOR_WHITE = 7

ncurses-9d.cpp

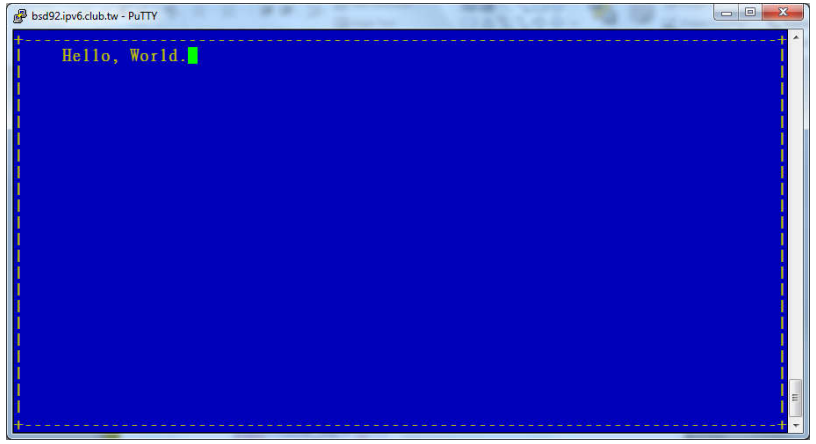
```
#include <curses.h>

int main()
{
        int j;
        initscr();
        start_color();
        init_pair( 1, COLOR_WHITE, COLOR_RED );
        init_pair( 2, COLOR_WHITE, COLOR_GREEN );
        init_pair( 3, COLOR_WHITE, COLOR_BLUE );

        for (int i = 0; i < 36; i++)
        {
            j = i % 3 + 1;
            attron( COLOR_PAIR(j) );
            printw("%2d", i);
            if (i % 6 == 5)
                printw("\n");
            attroff( COLOR_PAIR(j) );
        }

        refresh();
        getch();
        endwin();
        return 0;
}
```

COLOR_PAIR(0) defaults to foreground WHITE and background BLACK. It cannot be changed.



Change background color for the whole screen

```
// ncurses-9c.cpp
#include <curses.h>

int main()
{
        initscr();
        start_color();
        init_pair(1,COLOR_YELLOW,COLOR_BLUE);
        init_pair(2,COLOR_BLUE,COLOR_YELLOW);
        init_pair(3,COLOR_BLUE,COLOR_WHITE);
        // box(stdscr, '*', '-');
        border('|', '|', '-', '-', '+', '+', '+', '+');
        move(1,5);
        printw("Hello, World.");
        for (int i=1; i<=3; i++)
        {
            bkgd(COLOR_PAIR(i));
            refresh();
            getch();
        }
        endwin();
}
```



Create ncurses borders

- `int border(chtype ls, chtype rs, chtype ts, chtype bs, chtype tl, chtype tr, chtype bl, chtype br);`
  - ls - left side, rs - right side,
  - ts - top side, bs - bottom side,
  - tl - top left-hand corner, tr - top right-hand corner,
  - bl - bottom left-hand corner, br - bottom right-hand corner.
- FAQ: Can I draw a box around a small rectangle, instead of the whole screen?
- FAQ: Can I change the color for only a small area, instead of the whole screen?
- Ans: Of course. In that case, you are working with a "window" instead of *stdscr*.
  - See the manpages of subwin(), delwin().

Exercise: Arrow Keys

- Modify ncurses-6.cpp so that users can use arrow keys and HJKL to control the movement of ‘Q’.
- Moreover, try to allow users to use both uppercase ‘H’ and lowercase ‘h’ to do the same movement.
- Users can also change the color of ‘Q’ by pressing ‘0’…’7’.

int timeout(int delay)

- The **timeout**() function affects the behaviour of **getch**() when reading a character from stdscr.
  - If delay is a positive number, then **getch**() will wait for that many milliseconds before returning
  - If no character was available, then ERR (-1) will be returned.

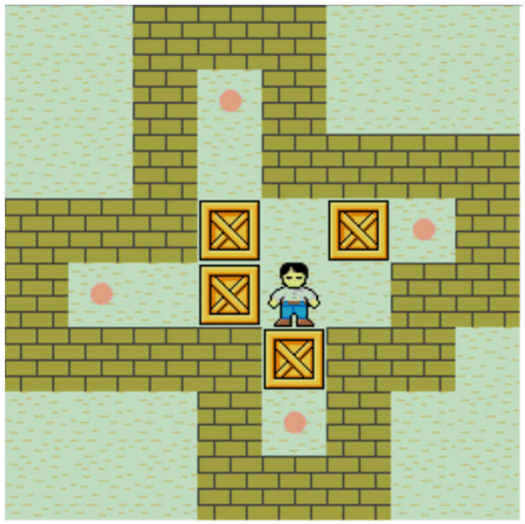
ncurses-10.cpp

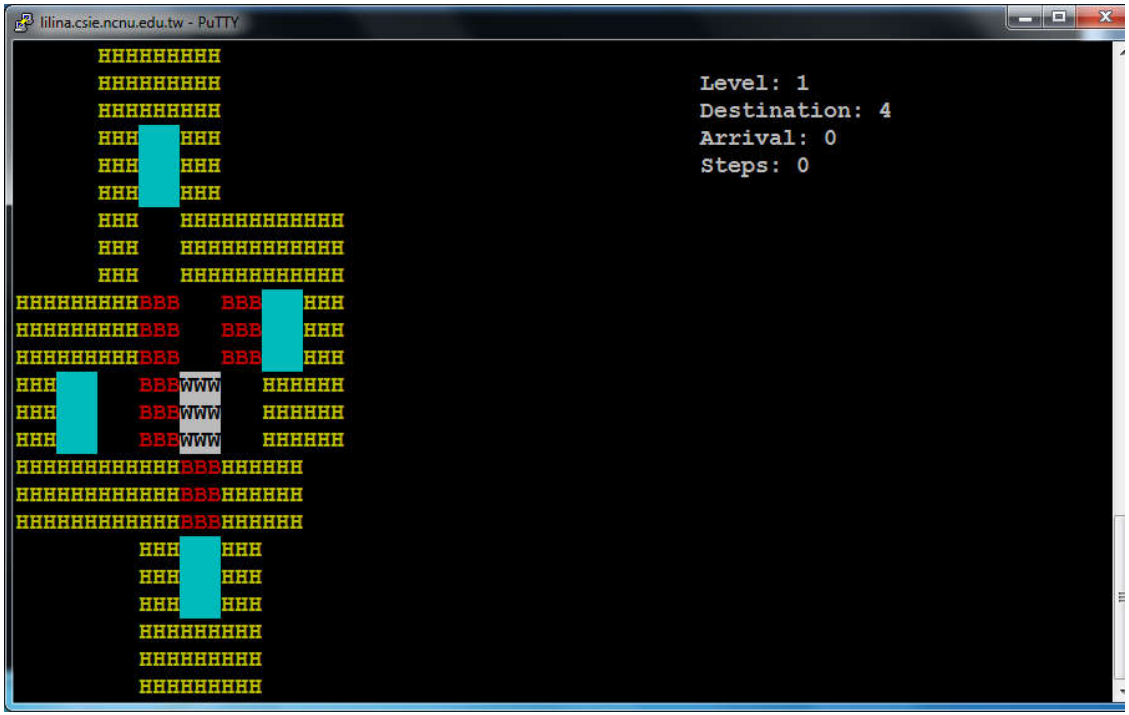
```
#include <curses.h>

int main()
{
    int x=10, old_x;
    int c;
    initscr();
    move(15, 15); addch('T');
    noecho();
    curs_set(0);// hide the cursor
    timeout(500);
// getch() will only wait 0.5 second

    for (int y=1; y<15; y++)
    {
        move(y, x); addch('Q');
        refresh();
        old_x = x;
        c = getch();
        switch(c)
        {
        case 'h':
            x--;
            break;
        case 'l':
            x++;
            break;
        }
        move(y, old_x); addch(' ');
    }

    refresh();
    endwin();
    return 0;
}
```

Exercise: `timeout()`

- Modify your previous exercise about "Show Time" so that
  - You can move the current time with arrow keys.
  - You can change the color of the time by pressing 1 ~ 7.
  - Even if you don't press any key, the current time keeps being updated.
    - Run "showTime-2.exe" to have a feeling about how your program should behave.

HW: Sokoban




lilina.csie.ncnu.edu.tw - PuTTY
Level: 1
Destination: 4
Arrival: 0
Steps: 0

Please come to approach me after class

- 111352004 洪彥宇
- 111352020 王培驊
- 111352024 邱冠銘
- 111352025 黃光岳
- 111321076 李佳蓉
- 112321033 尹淑曼