

# Introduction to Programming II

## Curses

Chris Brooks

Department of Computer Science  
University of San Francisco

Department of Computer Science — University of San Francisco — p. 1/77

### 24-2: Curses

- Curses is a C library for displaying ASCII characters at arbitrary points within a text window.
- Nice for making better-looking text-based UIs.
- C also has a number of third-party GUI libraries
  - Nothing standard like Java has

Department of Computer Science — University of San Francisco — p. 2/77

### 24-3: Hello world in curses

```
#include <curses.h>
int main(void) {
    int c;
    initscr();
    move(10,25);
    printw("Hello world!");
    refresh();
    c = getch();
    endwin();
    return (0);
}
```

Department of Computer Science — University of San Francisco — p. 3/77

### 24-4: The pieces

- `#include <curses.h>`
- `initscr()` - this sets up the curses window. It should be the first line in your program.
- `refresh()` - redraws the screen.
- `endwin()` - this cleans up curses. It should be the last thing in your program.

Department of Computer Science — University of San Francisco — p. 4/77

### 24-5: Placing the cursor

- curses uses (row, col) as its indexing method.
- (0,0) is the upper-left corner.
  - In other words, (y,x) is the representation
- To place the cursor at a particular location, do `move(y,x)`.

Department of Computer Science — University of San Francisco — p. 5/77

### 24-6: Output

- `printw()` - a printf-style function that places a string at the current cursor location.
- `putch()` - write one character at the current location.
- And, if you're feeling lazy, you can do:
  - `mvaddch(row,col, char)`
  - `mvprintw(row,col, string, arg1, arg2, ...)`
  - to combine moving and writing.

Department of Computer Science — University of San Francisco — p. 6/77

#### 24-7: Example - adding a marquee

- Let's add a border to our "Hello World" app.

#### 24-8: Example - adding a marquee

- Let's add a border to our "Hello World" app.
- Actually, we could also use the `box(stdscr, hchar, vchar)` function.
- Notice that `box` takes three arguments: window name, horizontal char, vertical char.
- We can actually have multiple 'windows'.

#### 24-9: Getting input

- We can read input with:
  - `getch()` = works just like `getc()`
  - `scanw()` - works just like `scanf`.
- In this case, we use `getch()` to leave the input on the screen.
- We can also use `usleep()` to specify a number of microseconds the program should sleep.

#### 24-10: Animation

- To change the appearance on the screen, we just draw over the old content and draw our new content.
- In curses, that means writing spaces over areas we want to be blank.
- How can we make the print statement crawl across the screen?
- Let's modify it to print a command line argument.

#### 24-11: Text display

- We can also set the attributes of the text that's printed, using `attron()` and `attroff()`
  - Interesting attributes are:
    - `A_NORMAL`
    - `A_STANDOUT`
    - `A_UNDERLINE`
    - `A_REVERSE`
    - `A_BLINK`
    - `A_INVIS`

#### 24-12: Exercise

- : Make the 'hello' string crawl left to right, with reverse video text.

#### 24-13: Dealing with input

- As we mentioned before, you can use `scanw()` and `getch()` to read in input.
- Let's use this to create a simple quiz program.

#### 24-14: Clearing the screen

- We can use `clear()` to clear the whole screen.
- Let's add this to our quiz program. Remove the question before printing 'correct' or 'incorrect'.

#### 24-15: Using Curses in Project 4

- Where do you need to use this in project 4?
- `drawBoard` should draw each component of the `boardArray`.
- Draw a '1' if the square is alive.
- Draw a '.' if the square is dead.
- Refresh once all cells are drawn.

#### 24-16: Advanced stuff: color

- To set the screen to draw in other colors, we do the following:
  - Add `start_color()` to the beginning of your program.
  - Create a set of color pairs - these indicate the foreground and background colors.
  - `init_pair(1, COLOR_RED, COLOR_BLACK);`
  - To start using a particular pair, set it as an attribute:
  - `attron(COLOR_PAIR(1));`

#### 24-17: Advanced stuff: color

- Curses has 8 builtin colors:
- `COLOR_BLACK, COLOR_RED, COLOR_GREEN, COLOR_YELLOW, COLOR_BLUE, COLOR_MAGENTA, COLOR_CYAN, COLOR_WHITE`
- You can also change the builtin colors by setting their RGB values:
  - `init_color(COLOR_RED, 700, 700, 0);`
- Modify the marquee program to use at least three different color pairs.

#### 24-18: Advanced stuff: dealing with arrow keys

- If you're making a game, you typically want the user to provide input via the arrow keys.
- The numerical values returned by the arrow keys are: 65,66,67,68.
- You can do:

```
int ch;
ch = getch();
if (ch == KEY_LEFT) {
    etc
}
```

#### 24-19: Exercise

- Let's use this to make a program that moves a character around the screen.
- When the user pushes 'up', move the character up one row. Etc.
- Issues:
  - Don't forget to undraw the old character.
  - Use `noecho()` to keep user input from showing up.
  - Move the cursor 'one step back' to place it on top of the character

#### 24-20: More stuff

- Other things we haven't touched on:
  - Curses also lets you create multiple windows. This is useful when you're making lots of changes, or you need to be able to revert back to a previous screen.
  - You can also use this to break the screen into multiple subscreens. This is useful if, for example, you want to have a menu at the top.