

ncurses in your hobostove

Eric Oestrich

B'more on Rails

2013-07-09



Eric Oestrich

SmartLogic Solutions

<http://www.smartlogicsolutions.com>

twitter.com/ericoestrich

github.com/oestrich



DISCLAIMER

I don't know ncurses
very well

This could be the worst
way to do things

Let's begin

```
gem install ncurses-ruby
```

Basic Setup

```
Ncurses.initscr
```

```
Ncurses.cbreak
```

```
Ncurses.noecho
```

```
# window initialization
```

```
Ncurses.doupdate
```


Basic Teardown

```
Ncurses .echo
```

```
Ncurses .endwin
```

Let me repeat that

Basic Teardown

```
Ncurses .echo
```

```
Ncurses .endwin
```

If you don't exit cleanly

Your command prompt won't be
happy

Learn to love reset

bin/hobostove || reset

Basic Panels

Panels - Setup

```
@win = Ncurses.newwin(  
    height, width,  
    starty, startx #it's backwards  
)  
Ncurses.box(@win, 0, 0)  
@panel =  
    Ncurses::Panel.new_panel(@win)  
# refresh
```

Panels - Refresh

```
# setup panels
```

```
Ncurses::Panel.update_panels
```

```
Ncurses.doupdate
```


Panels - Output

```
Ncurses.werase(@win)
```

```
@win.mvaddstr(y, x, "Hello")
```

```
Ncurses.box(@win, 0, 0)
```

```
# refresh
```

Re-implementing readline

or I didn't look to see if I could use
it in hobostove

(you can't)

User Input

```
while (ch = Ncurses.getch)
  case ch
  when 10 # enter
    message = ""
  else
    message = message + ch.chr.to_s
  end
  # move cursor and display message
end
```

User Input - Move cursor

```
@win.mvaddstr(1, 2, message)
```

```
Ncurses.move(  
    Ncurses.LINES - 2,  
    message.size + 2  
)
```

Abstract into Panels

- Handle most of this by itself
- Subclass for different panel types
 - InputPanel
 - UserPanel

Scrolling

[Github](#)

Trip ups

Modifier Keys

```
while (ch = Ncurses.getch)
  case ch
  when 14 # C+n
  when 10 # enter and C+j
  end
end
```


Bonus: GLI

DSL for creating command line
apps

GLI

```
require 'gli'
```

```
include GLI::App
```

```
program_desc "A command line  
Campfire client"
```

```
# commands
```

GLI

```
# set up
desc "Start hobostove"
command :start do |c|
  c.action do
    # stuff
  end
end
default_command :start

exit run(ARGV)
```

What it gives you

```
eric@bashir:[~/prog/src/hobostove (master)]$ bin/hobostove help
NAME
  hobostove - A command line Campfire client

SYNOPSIS
  hobostove [global options] command [command options] [arguments...]

GLOBAL OPTIONS
  --help - Show this message

COMMANDS
  help - Shows a list of commands or help for one command
  start - Start hobostove
```

Questions?

<http://www.smartlogicsolutions.com>

<http://www.twitter.com/smartlogic>

<http://www.github.com/smartlogic>

<http://www.facebook.com/smartlogic>