COM S 327, Spring 2018
Programming Project 1.05
User Interface with Ncurses

Last week we added some characters, and made them move around and smite one another. You may have added some code to drive your `@`. You can rip that code out, now[1]. We're going to add a user interface that you can use to drive your `@` manually. If you like, you can leave the auto-drive code in there and add a command to turn it on and off at runtime.

Still working in C, link in the ncurses library and use it for unbuffered I/O.

We're going to add stairs now, too. When the PC goes up or down a staircase, a new dungeon is generated and populated with the PC and new monsters. An upward staircase is represented with $<$; a downward staircase with $>$. The PC uses stairs by entering the appropriate stair command while standing on the staircase. NPCs cannot use stairs, not even the smart ones. Stairs provide an important means of escape for PCs. Each dungeon level should include at least one up and one down staircase located in rooms or corridors.

It's up to you whether levels persist or disappear. In most roguelike games, every staircase leads to a new, random dungeon (e.g., if you go down then back up, you will not return to the place that you just left). Persistant levels (down and up *does* return you to your old position) are harder to implement than non-persistant, but not significantly so.

It's also up to you whether or not you implement "connected stairs". That is, if you go down a staircase, are you left standing on an up staircase?

As for user input, all commands are to be activated immediately upon key-press. There is never a need to hit enter. Any command which is not explicitly defined is a no-op. Implement the following commands:

Key(s)	Action
`7` or `y`	Attempt to move PC one cell to the upper left.
`8` or `k`	Attempt to move PC one cell up.
`9` or `u`	Attempt to move PC one cell to the upper right.
`6` or `l`	Attempt to move PC one cell to the right.
`3` or `n`	Attempt to move PC one cell to the lower right.
`2` or `j`	Attempt to move PC one cell down.
`1` or `b`	Attempt to move PC one cell to the lower left.
`4` or `h`	Attempt to move PC one cell to the left.
$>$	Attempt to go down stairs. Works only if standing on down staircase.
$<$	Attempt to go up stairs. Works only if standing on up staircase.
`5` or space	Rest for a turn. NPCs still move.
`m`	Display a list of monsters in the dungeon, with their symbol and position relative to the PC (e.g.: "c, 2 north and 14 west").
up arrow	When displaying monster list, if entire list does not fit in screen and not currently at top of list, scroll list up.
down arrow	When displaying monster list, if entire list does not fit in screen and not currently at bottom of list, scroll list down.
escape	When displaying monster list, return to character control.
`Q`	Quit the game.

[1] If you gave your PC any special powers, like the ability to tunnel through walls, those should no longer apply, either.

With these changes, we no longer need the delay that we built in last week; the game will now pause automatically for input. And ncurses should handle the redrawing, so we're no longer spewing the entire dungeon to the terminal each turn. Things will look much nicer.

Our dungeons fill 21 out of 24 lines in a terminal. Display them on lines 1–21 (zero indexed). The top line is for message display. Use it to display any messages you like (like debugging information!). The bottom 2 lines are for status information, which we'll deal with in a later assignment.

You may add any other commands that you like, or map the required commands to additional keys, as long as you implement the specified mappings. If you choose to add other commands or mappings, please be aware that future assignments will specify additional commands and you may need to change your mappings. Future required commands will be limited to lowercase letters; therefore, you may choose uppercase letters, punctuation, control- and alt- modified keys, etc., without danger of future assignments forcing a change.