CIS*2500: Assignment 2

Due March 4, 9:00 AM

The Big Picture

You will write a program that draws a single room for a ″Rogue-like″ computer game. The program will parse a line of input text from an input file (room.txt), create the string for the room and its contents, and draw the room. Your program should also allow the user to move the hero around the room and pick up the treasure.

Starting

- Play Rogue for a half hour so that you get the idea of what you are building. There are lots of free rogue games online.
- The string tokenizer (strtok) will be extremely useful for this assignment. Google has lots of good info about how to use it.
- Use git to get the assignment skeleton that Judi has set up. It has all the folders you need and some empty files that you can edit
- You will need to add more files, the skeleton is incomplete this time
- be sure to use *git add* each time you add a new file
- Write a program to read a line from a file and break the line up into a char * array

Musts: do all of these to get a mark greater than zero

- Your solution must take the name of an input file as a parameter and use that file to draw the room.
- Your solution must use the character set from Rogue 3.6.3
- Your solution must compile without errors or warnings, must run without crashing or segfaulting
- Your soution must draw at least the outline of the first room in the test file

- Have separate source files for groups of tasks that repeat, such as dealing with operations for the room, parsing, drawing, and reading the file. Have a separate file for for main
- Have a README file and a makefile

Expectations for your game

- Correctly draw the room and all of the contents and doors
- Allow the hero to move around the room using the h, j, k, l keys.
- Permit the hero to pick up treasure, weapons and potions by walking over them. Print the list of things the hero has picked up when the program exits.
- Prevent the hero from walking through walls or over monsters. (no need to implement any sort of combat with monsters though)
- Allow the hero to walk onto doors, but not through them and doors should not be 'picked up'.
- Provide a way for the user to indicate readiness to move to the next room (there will be 9 rooms in the test file)

Expectations for your code

- Use the ncurses library
- Submission folder has src/ include/ and bin/ subdirectories that are used properly
- Source code is divided properly into .c and .h files
- Each .c file has the required header (see the policies document)
- Source code is properly indented
- Comments about function input/output and purpose are in .h files
- Comments about algorithm logic are in .c files
- Variable and function names are meaningful and are in camelCase
- A plain text file called README is in the root folder. It contains information about running and using your program as well as any known limitations of the program.
- You have a makefile that will compile your code and place the executable in the bin folder.
- Your code must compile with no error or warning messages using the -ansi and -Wall flags in gcc

Fun, possible bonus marks, completely unecessary

- write an algorithm to connect rooms with hallways and draw more than one room at a time (9 rooms max)
- implement rudimentary combat with monsters
- allow monsters to move randomly within the room

Details about Input String

- The input string will consist of a space-delimited line of text, no longer than 150 characters.
- The input string will not have a space at the beginning but may have trailing spaces and may have more than one space between elements.
- The input string will always have a newline at the end.
- The first element of the input string will ALWAYS be the room dimensions give as NumberXNumber (i.e. 10X12)
- The other elements can be in any order
- Door elements will begin with a lowercase d followed by a letter representing the wall that the door is in, followed by a position measured from north or west. e.g. de3 is a door in the east wall in position three from the north edge (where the corner is position 0).
- All other elements will begin with a lowercase letter followed by coordinates. i.e. the hero is represented by h6,8 which means that the hero is at position 6, 8 where the north west corner is position 0,0 (horizontal dimension given first).
- letters that can occur in element descriptions will be limited to t (treasure), h(hero), p(potion), w(weapon) and m(monster)
- 10X16 de4 dw9 ds8 t8,7 m3,4 h6,5 is one possible example of a room description string
- there will never be more than one door in a wall
- You need do only minimal error checking. The input file used for grading will not contain intentional errors.

Deliverables: Things you must hand in

1. (via git) your A2 folder with all required subfolders, c files and header files, README and Makefile
2. (via bucky) your collaboration statement
3. your test file with the data you used to test your program

Notes

1. Assignments that do not compile will be given a grade of zero
2. Assignments that compile with warnings may be given a grade of zero
3. Assignments will be compiled and marked using a linux terminal (same configuration as the Thornborough lab). Your code must compile and run correctly on that configuration.
4. All work in this course is to be done independently. Submissions will be examined electronically for similarity.