CIS*2500 Intermediate Programming (Winter 2015)

NAVIGATION

Home
- My home
  - My profile
  - Current course
    - CIS*2500_W15
      - Participants
      - Badges
      - Weeks 4-6 (Q2)
        - **Assignment Two**
  - My courses

ADMINISTRATION

Course administration

My profile settings

Assignment Two

CIS*2500: Assignment 2

Due February 13, 11:55 pm (git time)

Final Rev: (change log)

- added pd to the command list
- moved setxy, st, ht, clean, and print to enhancements section
- specified logo colours to be used as blue, green, red, yellow
- assignment will be graded without appointments during reading week

The Problem

Write an interpreter for a small subset of the Logo Programming Language (http://www.cs.berkeley.edu/~bh/logo). Your interpreter must take user input one line at a time, produce the correct output, provide logo-style error messages if the user provides incorrect input, allow the user to begin a new program, and save the existing program to file.

The user interface must have a drawing viewer, an input space, and a place for messages but can have additional features. You may not require the use of a mouse. The user input must be entirely from the keyboard.

Musts: We wont grade it unless you meet this standard

- Compile, with no errors using -std=c99 -Wall -pedantic(on the raspberry pi raspbian
- Run and draw the environment without crashing.

- Uses ncurses
- Allow the user to enter logo commands.

The Basics: The first 80%

You can get 80% on this assignment by just following the instructions. If you meet each of the expectations listed, you should get most or all of the 80 marks allocated to basic functionality.

Expectations for your Game

- The entire workspace should be no larger than 100x100.
- A logo command is executed when the enter key is pressed.
- Logo style error messages are provided in the case of errors
- The program is robust with respect to input and does not crash with incorrect input.
- The user is prevented from moving the cursor to inappropriate areas of the workspace
- Turns commands (rt and lt) can be given for 90, 45 and 30 degree angles. Differing angles will not be combined during testing.
- The user interface is intuitive and easy to use and contains the drawing, an input, and a message space as a minimum.
- The following logo commands are possible: fd, pu, pd, rt, bk, lt, home, setpencolor (using logo colours blue, green, red, yellow)
- User input is taken from the keyboard.
- User can save the program to file.
- User can begin a new program without restarting the software.

Expectations for your Code

- Your solution makes use of structs, pointers and arrays
- Your solution correctly uses dynamic memory
- Your solution uses a text file to save the logo program in the assets/ subdirectory
- Your solution uses functions effectively.
- Header files (.h files) contain function prototypes, struct definitions, typedefs as well as comments for each function.
- Each logical group of functions is in a separate .c file. Main is in a separate .c file

Organization and Coding Style (will lose marks if you dont meet these)

- Root assignment folder has src/ include/ bin/ and assets/ subdirectories

that are used properly
- Each .c file has the required academic integrity statement (see the policies document)
- Source code is properly formatted (will be checked with cpplint) and appropriately commented
- Variable and function names are meaningful and are in camelCase
- A makefile must be submitted that compiles your code with -Wall -std=c99 -pedantic. The binary executable should be called runMe.

Enhancements: How to get the last 20%

The final 20% of the marks for this assignment are earned by showing that you have mastered the programming constructs and that you are an adept problem solver.

You do not need to do all of the following to get graded on the last 20% of the assignment, but you do need to demonstrate (verbally when we grade you) that you have a solid understanding of how to approach them all and you need to have done at least 50% of the work described by these enhancements. Some of them are harder than others.

- Load a program from file and run it
- implement setxy, st, ht, clean, and print
- implement repeat
- implement procedures (but not recursive ones)
- implement logical operators
- implement arithmetic
- implement if/else
- implement iftrue
- implement while or do until(or both)
- implement an approximation of any degree of turn (rather than just 90/45/30).

Deliverables: Things you must hand in

1. (via git) Your entire submission must be in your A2 repository in git including the README file and a makefile.
2. Tag your submission with a2final. Use exactly that tag including case. You can delete the tag and reuse it if you update your code after tagging.

Grading

~~This assignment will be graded in person by appointment. Failure to make an appointment may result in grade penalties of up to 5%~~.

This assignment will be graded during reading week without appointments. This change is necessary to avoid falling too far behind in grading.

- Programming Style: Programming style will be determined by running cpplint on your submission and deducting one point for each error or warning. TAs will give no marks for programming style, they will only deduct marks for errors. The version of cpplint we will use is in the files folder on the course website. You may check your own code against it as you code.
- Submission Organization Have you organized your submission correctly into subfolders? Do you have a readme and makefile? Are they correct? Do you have the academic integrity header? You get no marks for submission organization, you can only lose marks for failing to follow the instructions.
- Program Operation(5) Input/Output, User proofing, Required UI sections.
- Functionality(7) Does it work properly? How many of the logo commands are implemented? Is the UI usable?
- Programming Constructs(8) Use of structs, pointers, arrays, files and dynamic memory
- Enhancements(5) TAs will look at the enhancements you did. It is best to document your enhancements in your readme file in case your assignment gets graded without you present.

Submission status

Submission status	This assignment does not require you to submit anything online
Grading status	Not graded
Due date	Friday, 13 February 2015, 11:55 PM
Time remaining	7 days 7 hours
Last modified	Friday, 23 January 2015, 1:48 PM
Submission comments	▶ Comments (0)

You are logged in as Nicholas Major (Log out)

CIS*2500_W15