

ICS 161 – Winter, 2016 – Homework 5

Follow the steps below. Start in your Wednesday, Jan. 27 lab session, and continue on your own time. Upload your files to the EEE Dropbox “ICS161-Hw4” before 11:55pm on Wednesday, Feb. 3. Note that part “A” of this homework requires you to complete a EEE Quiz between 3:00pm and midnight on Monday, Feb. 1.

Part A – The Event Queue Pattern

The lecture on Monday, Feb. 1 will primarily be on the Event Queue pattern from Nystrom’s book. After the lecture, take a short EEE Quiz before midnight on Monday. The URL of the quiz will be mailed to the class mailing list, and this homework assignment will be updated with the URL.

Part 1 - Texty

In <http://frost.ics.uci.edu/ics161/ForHomework4.zip> you will find Texty.h, which defines a very simple text output class. A Texty object has a True Type font and size. With its write() methods, a Texty object can output text to the window (actually, to an SDL_Renderer) at a specific x, y location, at the same y location as last time with x set so the text follows the previous text, and on the next line, starting at the x value from the previous write with a specified x and y.

The zip file also contains a TextyDemo.cpp program which utilizes the Texty class. You can run it through the provided TextyDemo.exe file in the Release subdirectory; double click on it in Windows Explorer. Use the right arrow key to move the background and observe the change to the text. Note that despite what image.png (from Lesson 4) says, the only implemented way to exit is by clicking the X button.

The homework assignment is:

1. Code Texty.cpp, an implementation of the Texty class. You shouldn’t make any changes to the public part of Texty.h – if you really need to, make sure the motivation and change are well documented in Texty.h. You are welcome to make changes to the private part of Texty.h.
2. Your implementation should handle error conditions, such as not being able to find, open, and use the specified font, in a reasonable way. Add a comment to the top of your Texty.cpp file explaining how your code handles error conditions.
3. You may modify TextyDemo.cpp as you see fit. Your modified version should retain right arrow key functionality. Your Visual Studio project should only require the three files Texty.h, Texty.cpp, and TextyDemo.cpp.

Part 2 – Better Texty

Add one of the following features to Texty, and update TextyDemo to show off that feature. Comment your code to describe your feature and make it very clear what you have done in terms of code.

- A. Texty has no “memory” of previously written text. A use case would be when we want to write log or error messages to the screen, and want previous messages to still appear. Modify Texty to add capability for this. Bonus: scrolling!
- B. Texty’s text output is always black, which isn’t good when the background is black. Make the text “background aware” so that it can always be seen. (This is different than just using a color other than black.)
- C. Make Texty \n aware, so that the NEXT_LINE option is unnecessary. Revise the two write() methods accordingly.
- D. Another modification of your choice. Email Prof. Frost before the end of day on Sunday, Jan. 31, to get approval (or not). Do not suggest “text in a color specified by the programmer” or “change the font or font size for an existing Texty object” – these seem too easy.

Turn in a zip file that contains your source code, images, fonts, your TextyDemo.exe, and the DLLs required to run the exe. Make sure these are in the proper directory structure, so that we can decompress the zip file, double click on TextyDemo.exe, and the program will run. (Try this out before uploading the zip file.) If any code is incomplete or has known bugs, discuss this in a comment at the top of Texty.cpp.