

C++ Assignment 4 - Concurrency and IO – Due Friday March 29th at 11:59 PM.

Outline: in this assignment you will get familiar with C++'s IO tools and concurrency. Your goal is to create a Nifti image loader which is capable of asynchronously reading .nii files from disk and then displaying the image on an OpenGL texture.

Part 1: Nifti image loader (75%) Nifti files are image files ending in a .nii extension, commonly used to store 3d or 4d medical imaging data. I have provided you with a single 3d nifti image (example below) called t1.nii.gz (the .gz indicates that it is compressed) on moodle.



As you can see it is a 3d MRI image of a brain. Each panel shows an orthogonal 'slice' through the image (axial, sagittal, and coronal slices respectively from left to right).

Your task is to create a C++ program that can read this nifti image and then display it as a texture using opengl. You will need to read up on the nifti image specification (resources below) which gives the header format. Once you are able to read the header, you can use the information provided in the header to read the full image data and place it in a 3d array and then show a single axial slice (leftmost image above) on a texture using OpenGL. Your program should also check to see if the image is zipped (in .gz format) and if so, it should unzip the image before loading it.

<https://brainder.org/2012/09/23/the-nifti-file-format/>

<https://nifti.nimh.nih.gov/pub/dist/src/niftilib/nifti1.h>

Use C++'s std IO stream libraries to handle the image loading. You can also check github for examples of how to read the image data, for example [here](#) is a class representing a Nifti image header. You can do something similar (represent the header as its own C++ class, and the image itself as a separate class).

Part 2: Concurrency (25%): once you have succeeded in loading the image and displaying an axial slice on an opengl texture, refactor your code such that the image loading is done *asynchronously*. This means that your program will have two threads:

1. A main thread that is running the OpenGL window, handling user input (if any), etc.
2. A second thread that handles the unzipping and loading of the nifti image. This thread will terminate once the image has completed loading.

The purpose of using two threads is to allow the user to continue interacting with the program while the image is being loaded. For the purposes of this assignment, you can display a slowly rotating triangle in the opengl window that will be replaced by the axial slice once the image finishes loading (the rotating triangle is simply to show that your main thread is still active while the image loads in the background).

Feel free to add some extras such as showing all 3 views (axial, sagittal, and coronal), allowing the user to scroll through the slices using the mouse, or a button to load other images from the file system.