**fCS-255** Computer Graphics Assignment (only 1 assignment, worth 20% of module)
Date set : 27/1/2020;
Deadline : Monday 2nd March 2020 at 11:00
Viva booking: https://graphicsvivas2020.eventbrite.co.uk

By submitting this coursework, electronically and/or hardcopy, you state that you fully understand and are complying with the university's policy on Academic Integrity and Academic Misconduct. The policy can be found at https://myuni.swansea.ac.uk/academic-life/academic-misconduct.

Guidance: **A lot of guidance for this assignment will be given in the lectures and support will be given in the lab classes**. Email J.O.Whittle@swansea.ac.uk with any questions.

**Unfair Practice**: Do not copy code from colleagues, internet or other sources. You may discuss approaches together, but all coding must be your own. Presenting work other than your own at a viva is plagiarism and a recipe for disaster.

**Aims**

Understand how an image is stored internally, and how to manipulate the image
Translate useful graphics algorithms into working code
Improve your programming skills
Understand that graphics can be useful to users (in this case within the medical context)
Work with a three-dimensional data set
Combine interaction with visual feedback
Practice presenting your work in a viva situation

**Files**:

The supporting framework is written in Java. You may build on this framework. If you wish to carry out the coursework in a different language you may do so, but there will be no provided framework. The assignment is to be demonstrated in the departmental PC Lab (you may use your own laptop). You will be required to demonstrate your working program from the deadline onwards at times to be arranged by you following the link above. You will require several things to start the exercise:

1. A copy of the Java template – downloadable from blackboard. This demonstrates how to display and manipulate images, and also functions and procedures which will help you with the exercise.
2. A data set to operate on – CThead.

[Note: You should not redistribute this data set anywhere – you will not have permission to do that]

**Exercise**:

1. Implement the following [40%]:
   a. Display a slider to allow the user to move through slices arbitrarily (currently a function to display slice 76 is provided)
   b. Display front and side views in addition to the top view (with independent sliders for each view)
   c. Perform maximum intensity projection [note the sliders will not affect this image (see guidance lectures if this does not make sense)]. Make sure your ray goes all the way to the back of the data set. (i.e. **do not clip at 113** when it could go to 255).

2. Choose to implement **two** from the following list [30% each, maximum 60% for two]
   a. Display thumbnail images for all slices of the data set (use your own resize function, do not use example internet code that writes to a file then reads it back in again – this will be plagiarism, and is an awful way to do it anyway). Some solutions include having an array of images or creating a very large image and filling it with the thumbnails.

b. Allow the image to be resized – nearest neighbour sampling is a basic approach; bilinear sampling gets more marks (write your own code – don't use the java library/function to do the resize). Operate the resize using a slider.
c. Perform histogram equalization on the data set (not on the images) to scale from the signed short int range to the unsigned byte range. Do not use the image histogram equalisation function (from java library, or from my notes on image histogram equalisation). Do use the correct histogram equalisation that operates on the full source 3D data set as indicated and discussed in the lectures. Generally, students that copy the simple image histogram equalisation get zero here.
d. Render from a different view apart from the ones above (i.e. a view from 45°, which will require sub-voxel sampling, either nearest neighbour or trilinear sampling) (hard). **This is not *image* rotation. You will get zero marks for implementing image rotation.** This requires some (straightforward trigonometry, doing it in 3D using 4x4 matrices gives the best implementation, and is easier to program/understand if you are comfortable with 4x4 matrices.) See chapter "Three-dimensional Geometric and Modeling Transformations" in Hearn and Baker textbook (ch. 11 in second edition, ch. 9 in 4th edition). If you really want to impress – implement arcball rotation with mouse interaction. To be clear – rendering from a different view will achieve full marks. The additional suggestions of 4x4 matrices, and arcball are challenges with no attached marks.

**Requirements**:
Also see later. You will demonstrate your program working to me, Joss or a post-graduate at times to be arranged in the week of the deadline or shortly after. You therefore need to have a comment with your name and student number at the top of your java files along with the declaration that it is your own work (if you have several files, place them in a ZIP – **do not include the data set**). **The coursework is worth 20% of this module. There is only 1 coursework. It is marked at the viva.**

**Mark distribution**:
See over for a breakdown. If you don't do the viva you will receive zero.

**Plagiarism**:
It's so important, I'm going to say it again. Each year a student will try to present code they don't understand – don't be that student. A simple question like "Why did you do that" or "What does that do" should not stump you.

**Computer Graphics Assignment**
**Marking scheme**

Note, if you cannot answer questions about your code (or have limited understanding of it), the marks will be reduced (sometimes down to zero).

First part [40 marks]
1. Has a slider been implemented correctly and is it fully understood in the viva? [12 marks]
2. Can all three views be seen with independent sliders and is it fully understood in the viva? [12 marks]
3. Has the maximum intensity projection been implemented correctly and is it fully understood in the viva? [16 marks]

Understanding can be tested using questions like: What is the purpose of MIP, what is the algorithm for MIP, how is this implemented, how are the other parts implemented, what does this bit of code do? Deduct appropriate marks from 1-3 above if the student cannot describe their own code. **Just reading comments out is insufficient and marks should be deducted. All in code comments must be in English.**

Second part [60 marks] (only your best 2 count)
1. Thumbnails: working visually [15 marks], additional functionality (e.g. click on a thumbnail to see it in the large window, or a good approach to coding (being able to deal with arbitrary numbers of slices) [15 marks]
2. Image resize: working visually [15 marks], additional functionality (e.g. bilinear interpolation to get a smoother image rather than nearest neighbour lookup) [15 marks]
3. Histogram equalization: Calculating the histogram correctly [15 marks], applying histogram equalization correctly [15 marks]
4. Different view: Work out and implement the equation to loop through the data set at an angle [25 marks]. Include tri-linear interpolation [5 marks]
5. If the student has done at least 1 from 1-4 above, does the student answer a suitable set of questions about their choices? Deduct appropriate marks from 1-4 above if the student cannot describe their own code.

**Submission Procedure**:
Submit the assignment through blackboard before the deadline. Demonstrate/viva your assignment after the deadline at times to be notified.

The college policy for late submission will be used. The timestamp from Blackboard will be used. I will email students at their University account, so you must read this frequently during the term. You might not be able to demonstrate/viva if you submit late.

If you have extenuating circumstances (documentation must be provided), we will not have a problem with making alternative arrangements.

**Java and Bytes**
Java does not support the unsigned types (e.g. unsigned byte). Images store the red, green and blue values as unsigned bytes (0 to 255), therefore some problems can arise - e.g. a value of 255 can be interpreted as -1. In order to work around this problem, one needs to carry out a logical AND with a bit mask of 255 and correctly typecast the result to int. e.g. int red = image_byte[index]&255; The MIP function contains such an example.

**FAQ**
How do I submit through blackboard?
On blackboard, use the following menus (this might be out of date?):

Assignments->View/Complete Assignment: Graphics ->Attach local file ->Submit

I see the error: "Please enter a valid file"
Perhaps you have a space in the name. Please move your file to a top level directory (e.g. c:\) and try submitting. If you still get problems, please ask at the library helpdesk.

Why won't you accept an emailed file?
I will delete emailed submissions. I have many students enrolled on this course. Blackboard provides a simple to use interface for accessing files – in a few clicks I can have a directory with sub-directories of student numbers, with your submission file in each directory. We also have to store assignments for 2 years after you leave so having them all in Blackboard allows me to do this in a simple way. It also timestamps assignment submissions (and you should receive a receipt).

I'm not registered on blackboard for your course.
Send me your student number for me to add you.

I've submitted the wrong version.
You can submit multiple times – I will mark the last version.

I cannot be available at the University on the days set aside for the viva.
Of course, we are fully sympathetic for extenuating circumstances (documentation must be provided). The lab sessions are timetabled and known well in advance. You are able to pick your viva slot from the start of the course.

Feedback?
Feedback is provided at the demonstration/viva and very shortly **after all vivas are complete** in an email to your University number mail account.

I haven't done any of the code, but I've read all the notes, book chapter, etc., so I am an expert in the various algorithms. Can I do the viva?
No. At some point in the near future you will probably want to work for a computing company and will need to program. This assignment is all part of the programming experience you will get at University to put you in a position where you can do that work. You don't usually get handed a good salary for no work in the real world.