CS-255 Computer Graphics Assignment (only 1 assignment, worth 20% of module)
Date set : 25/1/2021;
Deadline : Wednesday 3rd March 2021 at 11:00
Viva booking: An Eventbrite link will be posted on Canvas closer to submission.

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 Thursday advisory classes**.

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. The application which you demonstrate must be the code submitted to Canvas. To demonstrate code which is different to that submitted will count as Academic Misconduct.

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 through self-study and a challenging assignment
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. You will be required to demonstrate your working program from the deadline onwards using Zoom screen share. You will require several things to start the exercise:

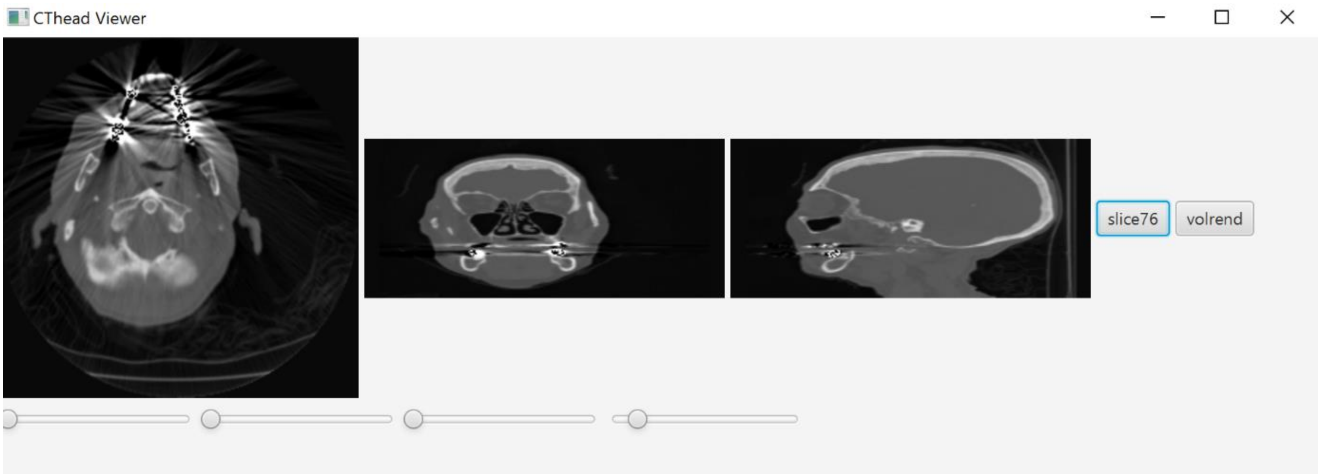
1. A copy of the Java template – downloadable from Canvas. This demonstrates how to display and manipulate images, and functions and procedures which will help you with the exercise.
2. A data set to operate on – CThead.

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

Exercise:

1. Implement the following [30%]:
    a. Display a slider to allow the user to move through slices arbitrarily (currently a function to display slice 76 is provided). **[10 marks]**
    b. Display front and side views in addition to the top view (with independent sliders for each view). **[20 marks]**

A minimal solution may look like this (you might have a better layout and different controls – the screenshot shows an extra slider and button used in Q2):



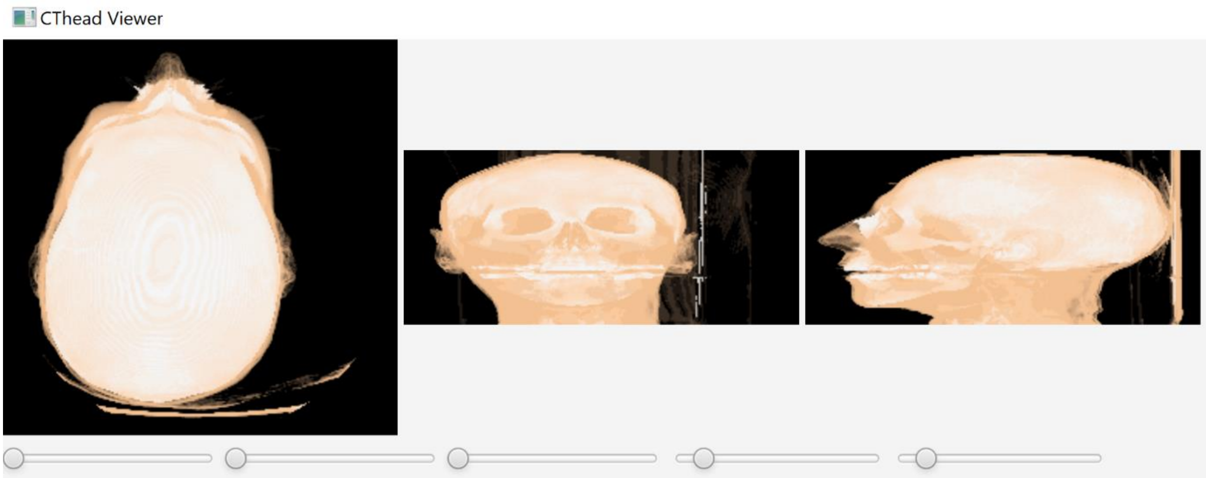
2. Perform volume rendering [50%]:
    a. You should use the transfer function (TF) defined as follows:
    CT values below -300 should be completely transparent.
    CT values from -300 to 49 should be RGB=(1.0, 0.79, 0.6) and opacity = 0.12.
    CT values from 50 to 299 should be completely transparent.
    CT values from 300 to 4096 should be RGB=(1.0, 1.0, 1.0) and opacity = 0.8.

    [40 marks]

    b. Link a slider to the opacity of the skin so you can move it between 0 (fully transparent) and 100 (fully opaque). Hint, the slider is int 0..100 but get the double value and divide by 100 to get a double value between 0 and 1 for opacity.

    [10 marks]

A minimal solution may look like this:



Note, the left image uses front-to-back volume rendering with the inner k loop going from 0 to CT_z_axis.

3. Do something advanced [20%]:
   This is to allow students to fulfil the descriptors of a 1st class student in the [QAA quality code] (see page 2, final column). Here I give you freedom to do something advanced and amazing. Note "consistently demonstrated exceptional initiative and personal responsibility" – we don't want to answer questions like "what is the minimum I have to do to get the marks?", or "will I get full marks if I do X?" (where X is not that tricky). Here are some suggestions:
   a. 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.
   b. When you find a non-transparent voxel, shade it according to a light source using diffuse shading. You will need lecture 15 Lighting from last year to see how to use the surface normal and a vector to the light source. You will need to calculate a surface normal from volume data. This is done using central differences (see the Derivatives and Finite Differences video from lecture 9 Edge detection). Briefly, a surface normal (Nx,Ny,Nz) in volume data can be computed at voxel (x,y,z) using Nx=value at x+1 minus value at x-1 and so on for Ny and Nz. Compute light vector (from x,y,z to wherever you've put the light) and then you can compute diffuse shading. If you also do sub-voxel sampling (3a.) and small step sizes through the volume, you can find the exact point you cross from air, soft tissue to bone (CT value =400) and shade it giving something like the images from my early [research work] or slide 3 of last year's Lecture 3.

(3.a or 3.b are examples that on their own would get the 20%)

Submission Requirements:
You will demonstrate your program working to me, Liam or a post-graduate at times to be arranged in the week of the deadline or shortly after (these will be in the Thursday lab slots). Submit your assignment through Canvas **by the deadline**. 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.**

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.

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).

1. [30 marks]
a. Has a slider been implemented correctly and is it fully understood in the viva? [10 marks]
b. Can all three views be seen with independent sliders and is it fully understood in the viva? [20 marks]
Understanding can be tested using questions like: How is this implemented, how are the other parts implemented, what does this bit of code do? Deduct appropriate marks from 1-2 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.**

2. [50 marks]
a. Has a transfer function been correctly implemented? E.g., it could be if..else statements, setting up an array (look up table) or some other way. [10 marks]
b. Has volume rendering been implemented for one view? E.g., do the loops used go through each voxel on a line and test it against the TF and accumulate the opacity and colour correctly? Is the colour correctly deposited in the image. [30 marks]
c. Are the additional two views coded correctly? [10 marks]

3. [20 marks]
I expect around 10% of students to do something great in this section. Some detailed hints about 3D views or better lighting are given in the question. Partial marks are to be awarded if some progress is made. E.g., perhaps sub-voxel sampling (tri-linear interpolation) is implemented correctly, or a ray-box intersection as progress towards 3D view, or finite differences as progress towards getting a normal for shading. Some students may look at other research – e.g., Levoy's original TF or 2D TFs or adaptive termination. Perhaps they have a nice user interface for drawing TFs. Anything interesting, advanced and working should attract marks.
Specifically, if students get their code to work with a different public domain medical data set of a different resolution, this can attract 5 marks. For these 5 marks, it must not involve duplicating lots of code with different sizes, but rather such that the dimensions of the data source appear just once as constants (or typed in when the file is read) and all other code works so it is clear that any sized data could be used subject to those initial user input or 3 constants. A different TF will need to be used to suit the data values in that data set.

Submission Procedure:
Submit the assignment through Canvas 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 Canvas 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.

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

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.