Sample take-home exam questions

Unless otherwise stated, all answers should refer your own ray tracer project. E.g. the question "describe the changes" should be answered by describing in a concrete way how your own ray tracer should be changed. You should not write new code or modify code, but you should indicate where in your program code need to be added or modified and what the additions or modifications should achieve. General discussions about how to implement changed functionality will *not* be accepted.

Transformation and composite objects

To facilitate modelling a 3D scene, we want to add functionality to rotate or move objects in the 3D space – individualy or several in one operation. You can achieve this using a local coordinate system for each object and transform between this and the global coordinate system when doing the ray tracing.

The attached excerpt of 3D Computer Graphics explains how to use a matrix to describe transformations of an object and how to translate between local and global coordinate systems.

1.1 Describe the changes you need to make to have composite objects – i.e. objects consisting of several other objects. (The point of this is to move and rotate the entire group in a single operation.)

1.2 Describe the changes you need to make to transform (move and rotate) objects.

Testing

2. Describe in general terms how to properly test code. Then describe more specifically how you would test the changes of questions 1.1 and 1.2.

Light buffers

When computing shadows in a ray tracer there will be a large number of hit point computations. Given N objects and M light sources, we will roughly get N*M hit point computations.

To reduce the number of hit point computations, we can use a "light buffer". This means that for each light source, we partition the 3D space in octants – 8 regions of space delimited by three orthogonal planes intersecting at the light source.

This is more easily understood in 2D space where we have 4 quadrants around the coordinate system origin delimited by the x- and y-axes. The octants are a generalisation of this idea to 3D-space with the light source in place of the coordinate system origin.

We will associate each octant with the objects in that octant. Any light ray from the light source will pass through only one of the octants so for the hit point computations we need only consider the objects in that octant. Thus on average, the number of hit point computations will be reduced by a factor 8.

3. Describe how your ray tracer can be modified to implement light buffers.

Sample written exam questions

The written exam is not compulsory. You can take if you aim for a 5 grade on the course, or to increase your chances of getting a 4 grade.

Portable data
You have two programs, running on different systems, MA and MB. The programs are going to communicate with each other by exchanging binary data (through a network or a file).

1 What (more than one!) problem do you have to consider? How can you handle these problems? Give proper motivations and clear examples.

ADT
You are going to implement a system to model an office. The office has movable walls (screens) so it is easy to create or modify rooms. You are interested in what rooms there are and how to move between them. Construct an ADT to represent the office.

2.1 What ADTs will you use. Motivate!
2.2 What primitives will your ADTs have. Motivate!
2.3 How do you represent data in your ADTs. Motivate!

Memory references
You have a program running on a modern processor.

3.1 How does the memory reference pattern affect the performance of the program? Motivate!
3.2 What memory reference pattern would you prefer? Motivate!
3.3 How can you change the memory reference pattern. Give clear examples.

Speed
You have a working program. You want to make it more efficient.

4.1 Describe how you undertake this task. What kind of tools do you use (if any). Motivate!
4.2 Do you need to sacrifice other qualities of the program to make it more efficient? Motivate and give examples.
4.3 Why should you use tools for this task?

Heaps
A data structure can be stack or heap allocated

5.1 How do you express in the program which allocation to use?
5.2 What are the advantages and disavantages with the two kinds of allocation?
5.3 How does the choice of allocation affect an ADT. Motivate! Give examples.

Memory handling
In C, you have to handle heap memory allocation yourself. You allocate memory when you need it and deallocate when you are done with it.

6.1 What does it mean that a program is "leaking memory"? How can you find the source of the

leak?

6.2 How does the manual memory allocation affect the program and ADTs? Is this an advantage or disadvantage?

6.3 Are there cases where you can avoid the effect mentioned in 6.2? Motivate!

Macros

Macros are often used to avoid the overhead of ´function calls.

7 What disadvantages do macros have? Can you avoid these disadvantages? Motivate and give examples!

Documentation

8.1 What is implicit documentation? Explain and give examples.

8.2 Why is it a good idea to extract documentation from a program rather than having separate external documents? Motivate!

8.3 Why is it important to document the programming process, i.e. the changes as such and not simply the end result? Motivate!