Written exam in EDA075

"Mobile Graphics"

Department of Computer Science, Lund University

Six assignments, each worth 1.0 points

Allowed aids: none

Tomas Akenine-Möller, 046-2228031

2006-10-20, 14–19

Remember to answer the questions as thoroughly as you can without diverging from the question. You shall strive after giving an as clear picture of your understanding as possible. Also, please write so other than yourself can read it.

1: Mobile graphics in general

a) [.3p] One big problem with graphics on mobile phones is that graphics uses lots of energy from the battery. Assume that someone invented batteries that are similar to today's batteries, except that they last 100× longer. This would solve part of the problem since we now could let the GPU use much more power. What is the remaining problem with such a battery?

b) [.4p] Make an illustration of a typical memory system (with all the different types of memory) of a mobile phone.

c) [.3p] The *rasterization equation* contains two factors called *depth complexity*, denoted d, and *overdraw*, denoted o. Explain what these two terms describe, and how they are related in theory.

2: Mobile Graphics APIs

a) [.3p] The hierarchy of the scene graph can be used for hierarchical modeling and animation. In order to speed up rendering, the hierarchy can also be used. How? This requires adding extra code (but you do not need to write down code for it).

b) [.3p] As we saw in assignment 2, adding more transforms to a scene graph can be bad for performance. Assume it was possible to distinguish between objects that are *static* and *dynamic*. Static objects are affected by the entire transform path above the object (as usual), however, we assume that this transform path will not change over time. How can such rendering of static objects be optimized?

c) [.3p] In an M3G scene graph, not all possible edges (connection between two nodes) are allowed. Loops is one type of edges that are not allowed. Imagine a scene graph that allowed loops. Be creative, and think of uses for that, and how would you imagine that it would be implemented?

d) [.1p] An extended version of OpenGL ES is used as the primary graphics API for the PlayStation III (which is a non-mobile game console). Why did they not decide to use OpenGL instead?

3: Culling
a) [.4p] Explain how Zmin-culling works, and what kind of modifications are needed to an architecture not capable of either Zmin-culling and Zmax-culling.

b) [.2p] Object culling using "occlusion queries" can reduce the amount of computations and bandwidth usage. How does that work?

c) [.4p] Assume that you have an architecture that can switch (per tile) from using Zmin-culling to Zmax-culling and vice versa, i.e., you cannot use Zmin-culling and Z-max culling at the same time. When (and why?) would it be optimal to switch from Zmin to Zmax? How can that be determined?

4: Edge functions and fixed-point math
a) [.4p] Assume we have a number of front facing, connected, non-overlapping triangles lying in a shared plane. For robust rasterization it is important that each pixel is written to at most one time. How can you solve that with edge functions (all cases must be handled)?

b) [.3p] How can edge functions be used to test whether a tile overlaps a triangle? Explain why this works.

c) [.3p] Take a look at the following code, which uses three floating-point numbers, `af`, `bf`, and `cf`:

```
int a=floatToFixed(af,4);
int b=floatToFixed(bf,5);
int c=floatToFixed(cf,2);
int d;
d=a*b+c;
```

Obviously, the programmer wants to compute the product between a and b, and then add c in fixed point. Change the last line of the code so that the intended result is obtained, and so that as few fractional bits as possible are used (how many is that?).

5: Depth and texture compression
a) [.3p] Compressing the depth buffer is one way of reducing bandwidth. In

contrast to texture compression, depth compression cannot be fixed rate. Explain how that is solved in a depth compression architecture.

b) [.2p] Argue why it is expected to be easier to compress a tile of depth values that a tile of color values.

c) [.5p] Fenney describes, in his article "Texture Compression using Low-Frequency Signal Modulation," how textures can be compressed down to 2 bits per pixel. Describe how decompression is done for that algorithm by showing how the bits are used in a compressed block, and how they are used during decompression.

6: Miscellaneous
a) [.3p] Explain the concept of unified shaders, and the advantages of them.

b) [.4p] Try to invent an efficient color buffer compression scheme.

c) [.3p] In texture decompressors, it is common to expand a number, which is represented using few bits, to a number using more bits. Assume that a 4-bit number, x, is to be expanded into an 8-bit number. How would you do that efficiently, and why is that the correct way to do it?

The end!

Please fill out the "Course Evaluation form, and hand it in anonymously.