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

2007-10-16, 8–13

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] The Apple iPhone, and many other mobile phones, have specialized graphics hardware in order to efficiently draw the user interface etc. However, all rendering could have been done in software using the CPU as well. What are the advantages of specialized graphics hardware?

b) [.2p] Two high-level APIs for Java for graphics on mobile devices are M3G and Mascot Capsule. Discuss advantages/disadvantages of these APIs.

c) [.2p] Just a few years back, it could easily be argued that the pixel quality had to be better on a mobile phone compared to a desktop PC. What is the situation today? What has happened?

d) [.3p] Texture caching obvisouly helps a lot in terms of reducing bandwidth. Two different texture filtering methods are *mipmapping* and *nearest neighbor*. Explain and argue which of these works best with texture caching.

2: Mobile Graphics APIs

a) [.3p] Mention three important differences between OpenGL and OpenGL ES.

b) [.3p] In M3g, Describe how you can create a simple scene graph using the Java class `Group`. Assume you have three different `Group`s called `sun`, `earth`, and `moon`, and make sure that the groups have the same hierarchical relationship as the real sun, earth, and moon.

c) [.4p] In M3G, you can define a set of connected triangles using a triangle strip with an index buffer. With respect to Figure 1, assume we have a vertex

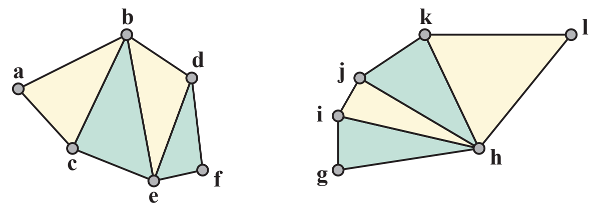

Figure 1: Two triangles strips with vertices.

buffer consisting of the following vertices and in the following order:
a, **b**, **c**, **d**, **e**, **f**, **g**, **h**, **i**, **j**, **k**, **l**.
In Figure 1, there are two groups of connected triangles. Create the index buffer for each of these, and make sure that only one triangle strip is used per group, and also that the strip is ordered correctly (frontfacing triangles have the vertices in counter-clockwise order). Use as few indices as possible.

3: Bandwidth
a) [.2p] Explain why it is more important to reduce bandwidth usage in a mobile device compared to a desktop PC.

b) [.2p] There are several differences between a texture compression/decompression algorithm, and buffer (e.g., color and depth) compression and decompression algorithms. Mention two important ones, and argue why these are important.

c) [.2p] In standard image compression (using for example JPEG), the involved algorithms are often *lossy*, and so are texture compression algorithms. Try to come up with arguments and/or solutions so that lossy algorithms can be used for color buffer compression.

d) [.4p] Make some reasonable assumptions about the number of bits per component of the depth buffer, color buffer, and textures. Assume that we are constructing a GPU with a target depth complexity of $d = 6$, and that we use only bilinear filtering but with two textures per pixel. We have already implemented a texture cache, with cache miss ratio $m = 0.25$. Which one of the units that use bandwidth to memory is in most need of improvement (assuming that performance is directly proportional to memory bandwidth usage)? Use theoretical models to motivate your answer.

4: Anti-Aliasing
a) [.2p] When determining the color of a pixel, we often have to use point sampling in order to get practical algorithms. How would we ideally solve the problem? Argue why that is hard.

b) [.3p] Mention three different types of aliasing in computer graphics.

c) [.3p] Assume for a while that the human is *only* sensitive to aliasing on horizontal edges. Construct a sampling scheme that helps reduce aliasing as much as possible in this case.

d) [.2p] Make drawings of the sampling positions for both RGSS and FLIPQUAD. Indicate (with numbers) exact positions of samples.

5: Edge Functions

a) [.4p] Derive a formula for an edge function from $\mathbf{a} = (a_x, a_y)$ to $\mathbf{b} = (b_x, b_y)$. It is not enough to simply print down the formula.

b) [.3p] Define barycentric coordinates, and explain how and why they can be computed with edge functions.

c) [.3p] If you have a set of connected triangles (sharing edges, and vertices) in the plane, a sample point should only be covered by one triangle. What happens if a sample point and a triangle vertex (shared by many triangles) coincides? Describe how the problem is sovled.

6: Miscellaneous

a) [.4p] Using a delay stream architecture, you can get better occlusion culling, solve order-independent transparency, and get adaptive antialiasing. Describe how order-independent transparency works when using a delay stream.

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

```
int a=floatToFixed(af,2);
int b=floatToFixed(bf,3);
int c=floatToFixed(cf,6);
int d=floatToFixed(df,2);
int e;
e=a*b+c*d;
```

Obviously, the programmer wants to compute the product between a and b, and then add product between c and d in fixed point. Change the last line of the code so that the intended result (without any loss of information) is obtained, and so that as few fractional bits as possible are used (how many is that?).

c) [.3p] 3D multi-view displays can have huge impact on mobile displays and for TV/video/computer displays in a near future. Explain how one type of display can convey a three-dimensional experience to the human observer, and discuss the costs of rendering to such a display.

The end