Exercises for Advanced Graphics (Lectures 5-8)

All work to be submitted by email in PDF format, no less than 48 hours before supervision.

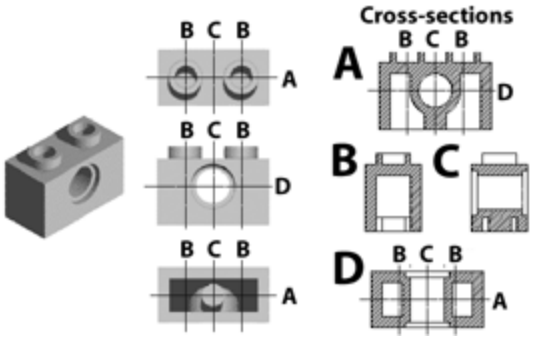
These exercises are partly drawn from past exam questions.

1. **Ray / Cone**
    a. Show how to find the first intersection between a ray and a finite-length, open-ended section of a right circular cone, apex at the origin, aligned along the x-axis, for which both ends of the finite length are on the positive x-axis ($0 \leqq x_{min} < x_{max}$).
    b. Extend this to cope with a closed cone (i.e. the same cone section, but truncated to a frustum with end caps). Take care to consider any special cases.
    c. Extend this further to give the normal vector at the intersection point.

2. **Ray / Polygon**
    Write an implementation of a ray / triangle intersection routine, using real code or pseudocode. Your method should return true if the ray hits the triangle, else false.
    a. Generalize your method for an arbitrary convex polygon.
    b. Generalize your method for an arbitrary convex polyhedron.
    c. How does knowing that the polyhedron is convex allow you to significantly optimize cases where the ray does *not hit* the polyhedron?

3. **OpenGL and Shaders**
    Using publicly available libraries and in the programming language of your choice, write a simple OpenGL program.
    a. Render a tetrahedron.
    Take screenshots as proof.
    b. Render a torus.
    Take screenshots as proof.
    c. Using shaders, implement an animated procedural texture on your torus.
    Record the animation and upload a ten-second (or longer) video showing the animated texture to YouTube.
    d. [Optional] Demonstrate your animated shader running on the Utah Teapot [http://people.sc.fsu.edu/~jburkardt/data/obj/teapot.obj] or Stanford Bunny [http://graphics.stanford.edu/~mdfisher/Data/Meshes/bunny.obj].

4. **Constructive Solid Geometry**
    a. List three ways of combining objects using constructive solid geometry (CSG), and describe how an object built using CSG can be represented using a binary tree.
    b. Given the intersection points of a ray with each primitive in the tree, explain how these points are passed up the tree by each type of combination node to produce a list of intersection points for the whole CSG object.
    c. Show how the Lego™ brick below can be constructed using Constructive Solid Geometry (CSG). You may assume the following primitives: sphere, cylinder, cone, torus, box.



5. **Radiosity**
   Explain form factors and view factors in radiosity. Outline an implementable method of calculating view factors. Describe how your method might leverage existing hardware acceleration.

6. **Photon Mapping**
   Explain the meaning of the term Monte Carlo integration and explain how it is demonstrated (twice) in the the Photon Mapping algorithm.