## Exercises for the second supervision

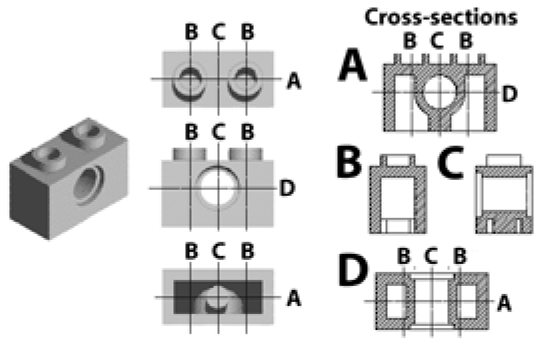
***All work must be submitted by email no less than 48 hours before supervision.***

These exercises are 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 cone, centred at the origin, aligned along the x-axis, for which both ends of the finite-length are on the positive x-axis (i.e. $0 < x_{min} < x_{max}$).
   b) Extend this to cope with a closed cone (i.e. the same cone, 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) **Curvature -** The *one-ring* of a vertex is the (usually ordered) set of vertices which lie exactly one edge away from a given vertex on a polyhedral surface. Given a vertex $V$ with one-ring $\{v_0,...,v_{n-1}\}$, give a formula for the discrete curvature of the surface at $V$.
   a) Bonus question: Given a hobbit with one-ring $\{v_0,...,v_{n-1}\}$, how would you rule them all?

3) **Constructive Solid Geometry**
   a) List the three ways of combining objects using constructive solid geometry (CSG).
   b) Describe how an object built using CSG can be represented using a binary tree.
   c) 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.
   d) 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. [You are expected to describe which primitives are needed and how they are combined but you are not expected to specify accurately all of the parameters of the primitives.]




4) **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.

5) **Photon mapping**
   a) Explain photon mapping, highlighting the two portions of the algorithm which show *Monte Carlo* integration.
   b) Explain how photon mapping produces *caustics*.

6) **Implicit Surfaces**
   a) Explain the special cases in the polygonalization of an octree, and how you might address them.
   b) Summarize the *marching cubes* algorithm.

7) **Shaders**
   a) Describe the inputs, outputs, and usage of *vertex shaders*
   b) Describe the inputs, outputs, and usage of *fragment shaders*

8) **Ray tracing** – A perfectly reflective mirrored sphere, S, is centered at the origin (0, 0, 0). Directly above it is a bright red $2 \times 2 \times 2$ cube, C, centred at (0, 5, 0). The default background colour of the scene is blue. A ray-tracing ray R is fired from (0, 1, 10) with direction (0, 0, –1). The scene is lit by an ambient light source and there are no other objects in the scene. What is the maximum radius of S such that the colour calculated for R is red?