Examen parcial de Sistemes Gràfics Interactius (SGI)

Curs 2012-13, 16 d'abril 2013

Name :

Question 1. Explain what an *ambiguous* 3D model is. Give an example of a representation scheme that might be ambiguous for representing solid models.

Question 2. Let A, B be two surfaces. We know that there exists a homeomorphism between A and B. What can we state about A and B?

Question 3. For some pointset S we know that S=cl(Int(S)), that is, S equals the closure of its interior. What can we say about the topology of S?

Question 4. Compute the genus of a solid object bounded by a single shell with 10 faces, 24 edges, 16 vertices and 2 rings.

Question 5. We want to write an efficient Python algorithm for counting the number of concave edges in a Blender mesh.

(a) Describe briefly the major steps of the algorithm. Indicate clearly which incidence relationship you need to build.

(b) Give the (average case) time complexity of the algorithm.

Question 6. A single piece of Emmental-like cheese such as the one on the figure has (select the correct choice):

(a) Single cell, many shells, large genus

(b) Many cells, single shell, large genus

(c) Many cells, single shell, small genus

(d) Single cell, many shells, small genus

Question 7. Which Boolean operation(s) will you choose to model a solid similar to the cheese object above using CSG?

Question 8. Explain how "face" vertices are computed in the Catmull-Clark subdivision scheme. Does the computation of "face" vertices require any additional topological relationship beyond those provided by Blender?

Question 9. For a valid polyhedron, indicate the cardinality of the following topological relationships:

(a) E:{F}

(b) E:{V}

Question 10. Indicate the minimum cardinality for the following topological relationships, assuming a valid BRep:

(a) V:{E}

(b) V:{F}

Question 11. Building efficiently the V:{F} relationship in Blender involves:

(a) Traversing all vertices (outer loop); for each vertex traverse the list of faces (inner loop)

(b) Traversing all faces (outer loop); for each face traverse its vertices (inner loop)

(c) Traversing all vertices (outer loop); for each vertex find its references in the list of faces

(d) Traversing all faces first (one loop) and after that traverse all vertices (second loop)

Question 12. Given the quantities F,E,V of a valid polyhedron, how can we compute the number of half-edges?

Question 13. One algorithm for creating BRep approximations of a sphere consists in applying a 1:4 subdivision to an initial triangular mesh approximating a sphere (e.g. an octahedron). Assuming the initial solid has F faces, E edges and V vertices, indicate the resulting number of faces F', edges E' and vertices V' after one step of 1-to-4 subdivision of its triangles.

Question 14. Given a polygon in 3D space, we know the areas (Sx, Sy, Sz) of its projections onto the X=0, Y=0 and Z=0 planes. How can we compute the area of the polygon from these data?

Question 15. The output of a Catmull-Clark subdivision step produces

(a) Faces with 3 or 4 vertices

(b) Only triangles

(c) Only quads

(d) Faces with arbitrary number of vertices

Question 16. Concerning subdivision surfaces (answer YES/NO)

(a) Is the Catmull-Clark surface guaranteed to go through the vertices of the input mesh?

(b) Does the Doo-Sabin surface interpolate the centroid of the input faces?

Question 17. The following list contains all the vertices forming face 1. Sort the vertices to form a face (starting from V3), following the domino algorithm.

V 3: xyz, {4,1,6}, fromAoutB
V 4: xyz, {2,1,6}, fromAoutB
V 7: xyz, {4,1,12}, fromAoutB
V 8: xyz, {5,1,2}, fromAoutB
V20: xyz, {8,1,5}, New
V22: xyz, {8,1,12}, New

V3,V4, …

Question 18. Regarding the marching cubes algorithm, draw an example of ambiguous face and an example of non-ambiguous face.

Question 19. Given two solids A, B, indicate which vertices will be part of the Boolean difference A-B.

Question 20. Write a Python function for Blender that, given a Mesh object **with a single shell**, and its genus H, prints "OK" if and only if the mesh satisfies the Euler equation.

```
def EulerCheck(mesh, H):
```