200 example questions for the CS3451 final

Jarek Rossignac

1 2D GEOMETRY

1.1 Points, vectors, dot product

1) Provide a formula for computing S(A,t,B), which moves linearly from A to B when parameter t varies from 0 to 1
2) Provide a formula for computing the dot product dot(U,V) and properties of dot(U,V): When is it >? When is it =0?
3) Provide a formula for a version R(V) of V=<x,y> rotated 90 degrees
4) Provide a formula for V2 and for |V|, where V=<x,y>
5) Provide a formula (using dot product) for isLeftTurn(A,B,C) testing whether polygon{..A,B,C..} turns left at B
6) Solve equation PQ•V=0 for t, with P=S+tT. Mention an application.
7) Provide the formula for the average of 3 points (A+B+C)/3 that does not use additions of points, but only proper operations on points and vectors

1.2 Lines, planes, edges, rays, intersections, reflections

8) Provide a parametric form of a point P(t) on edge(A,B)
9) Provide a simple test whether edge(A,B) intersects edge(C,D) (use dot product, not coordinates)
10) Provide a construction of the normal projection, project(P,A,B) of point P onto line(A,B)
11) Provide a test whether project(P,A,B), discussed in the question above, falls onto edge(A,B)
12) Provide a parametric form of line(A,B) through A and B
13) Provide an implicit equation of line(A,B) through A and B
14) Explain how to compute the intersection point X of line(A,B) with line(C,D)
15) Provide the parametric form of a point ray(S,T,t) of a ray from S with tangent T
16) Provide the code for testing whether ray(S,T,t) intersects edge(A,B), be careful that the ray does not go backwards
17) Computation of the parameter value t for the intersection of ray(S,T,t) with edge(A,B), assuming the intersection exists
18) Computation of the point X where ray(S,T,t) intersects edge(A,B), assuming the intersection exists

1.3 Triangles

19) Formula for the center of mass of triangle(A,B,C), where "triangle" means the surface area enclosed by 3 edges
20) Formula for the area of triangle(A,B,C)
21) Test whether triangle(A,B,C) is clockwise
22) Test, pmc(P,A,B,C), returning true when point P lies inside triangle(A,B,C)
23) Algorithm for testing whether edge(P,Q) intersects triangle(A,B,C)
24) Assuming pmc(P,A,B,C), how to compute the point X where ray(P,T,t) exits triangle(A,B,C)
25) Test whether triangle(A,B,C) and triangle(D,E,F) intersect (not just their boundaries)

1.4 Polyloops

26) Formula for estimating the tangent T at vertex B in polyloop {… A,B,C…}
27) Formula for estimating the normal N at vertex B in polyloop {… A,B,C…}
28) Displacement vector for vertex B in polyloop {… A,B,C…} during a step of the cubic Bezier subdivision
29) Displacement vector for midpoint of edge(B,C) in polyloop {… A,B,C,D…} during a step of the 4-point subdivision
30) Algorithm for performing one step of the Jarek subdivision J0.5
31) Algorithm for performing one step of a polyloop smoothing
32) Compare properties of the limit curves obtained by four-point, Jarek, or cubic B-spline subdivision
33) Difference between smoothing and subdivision
34) Algorithm for point-in-polygon test
35) Algorithm for computing the area surrounded by polyloop with vertices P[i], assuming n(i) gives the next index
36) Algorithm for evaluating a point P(t) on a cubic Bezier curve with control vertices A, B, C, and D

1.5 Distances

37) Explain the difference between the (minimum) distance and of the Hausdorff distance. Draw an example showing them both. Then suggest two important applications for each.

38) Formula for the distance, d(A,B) between points A=(xA,yA) and B=(xB,yB)

39) Provide an algorithm for computing the closest pair indices (m,n) in a set of sites (points) Pi in the plane

40) Mathematical formula for the distance d(S,T) between two arbitrary sets, S and T: d(S,T) = min….

41) Algorithm for computing d(S,T) when S is a set of points {Si} and T is a set of points {Tj}

42) Algorithm for computing the distance between edge(A,B) and edge(C,D)

43) Math formulae (2 versions) for the Hausdorff distance h(S,T) between two sets, S and T (using maxmax, using inflation)

44) Algorithm for computing h(S,T) when S is a set of points {Si} and T is a set of points {Tj}

45) Algorithm for computing the Hausdorff distance between edge(A,B) and edge(C,D)

46) Love story metaphor providing intuition for d(S,T) and h(S,T)

47) Example why we cannot compute the Hausdorff distance of two polyloops by only considering pairs of edges

1.6 Frames

48) Formula for the location of point P, given its local coordinates (x,y,z) in a given frame (O,I,J,K)

49) Expression of the local coordinates (x,y,z) of point P in a given frame (O,I,J,K)

50) Given A1, B1, C1, A2, and B2, computation of C2, such that triangle(A1,B1,C1) and triangle(A2,B2,C2) are similar

2 Animation

2.1 Motions

51) Equation of the trajectory P(t) of a point moving with constant velocity V and starting at position P

52) Computation of a point P(t) moving along a cubic path that starts at A with velocity U and ends at D with velocity V

53) Equation of the trajectory P(t) of a point starting at P with velocity V and subject to a constant acceleration G

54) Algorithm for animating (producing the frames at constant time steps) of the above free-fall trajectory

55) Algorithm for animating the interpolating motion of a stick between edge(A,B) and edge(B,C)

56) An artist has drawn a series of 6 edges (one per key frame). Your job is to produce a smooth animation showing the smooth evolution (motion, rotation, stretching) of an edge that evolves smoothly with time and interpolates the key frames. Approach 1 would interpolate the end points linearly between consecutive frames. Explain exactly how you would implement this and draw an example showing how it would work and illustrating why the result would not be smooth. Approach 2 would use a four-point subdivision to define the trajectory of the end-points. Explain how you would implement it and draw an example, which shows its advantages over approach 1, but also limitations. Explain the limitations. Explain what your approach 3 should accomplish to overcome these limitations and suggest how this could be implemented.

2.2 Deformation

57) We want to produce an interactive tool that makes it easy for the artist to produce a short smooth animation of a short smooth curve (that deforms through time). Explain what you would use to represent the animation, how the artist would be editing it, and how you would play the animation. You should use the cubic Bezier formulation for it. First explain how to evaluate a point P(t) on a cubic Bezier curve with 4 control points A, B, C, and D and provide the code bezier(A,B,C,D,t) that returns P(t). Then explain in details your data structure for representing the animation and provide the code for rendering the frame (smooth Bezier curve) of the animation for any given time t between 0 and 1.

2.3 Collision, shock

58) Compute when disk(C,r) with initial center C and radius r traveling with constant velocity V will collide with line(A,B)

59) Algorithm for computing the new velocity W of the disk after the above elastic collision, when the line is fixed

60) Compute when disk(C1,r1) traveling with constant velocity V will collide with disk(C2,r2)

61) Assuming that disk(C1,r1) and disk(C2,r2) are tangent, but not one in the other, what is their contact point X

62) Compute when disk(C1,r1) traveling with velocity V1 will collide with disk(C2,r2) traveling with velocity V2

63) Compute whether, and if so when, edge(A(t),B(t)) collides with point P, where A(t)=A+tU and B(t)=B+tV

64) Provide a simple test whether triangle(A,B,C) and triangle(D,E,F) intersect. Do not use edge/edge intersection tests.

3 Computational geometry

3.1 Convexity

65) Code for computing n(i) and p(i), which return the index to the next and previous vertex around the loop {Pi}

66) Test whether polyloop {Pi} is free from self-intersections

67) Test whether polyloop {Pi} is convex

68) Intuitive definition of a convex hull of a set of points {Si}

69) Can the convex hull of a polyloop be computed by iteratively removing concave vertices?

3.2 Delaunay / Voronoi duality

70) Define a Delaunay triangle, given a set of sites (points) Pi in the plane

71) Explain how to test whether triangle(Pi,Pj,Pk) is a Delaunay triangle

72) Define a Voronoi region, given a set of sites (points) Pi in the plane

73) Give examples of applications of the Delaunay triangles and Voronoi regions

74) Explain the duality between Delaunay triangles and Voronoi regions

75) Given a Delaunay triangulation, explain how to compute the Voronoi region of site Pi

4 Topology

4.1 Set operators

76) Set formulations of complement !S, interior iS, boundary bS, exterior eS, and closure cS of a set S

77) Math definitions of union A+B, intersection AB, and difference A–B between two sets A and B

78) Hatching a set defined by a Boolean (CSG) formula, such as A–(B–CD), given the sets A, B, C, D

79) Construction of a CSG formula for a desired set in 2D, given the primitive sets A, B, C, D

80) Algorithm for testing whether a point P is in a region represented in CSG

4.2 Interference

81) Test whether disk(C1,r1) and disk(C2,r2) interfere

82) Test whether circle(C1,r1) and circle(C2,r2) interfere

83) Algorithm for testing whether two polygons (connected regions bounded by straight edges) interfere

4.3 Boundary representation of 2D regions

84) We use a set of polyloops to represent a region R. They may cross themselves and each other. Provide a definition for R, which matches the intuitive one in cases when the polyloops are intersection free, and an algorithm for testing whether a point Q is in R.

5 Rendering

5.1 Ray tracing in 2D

85) Parametric form of point P(t) on ray(V,T) from viewpoint V in the direction T

86) Algorithm for computing the testing whether ray(V,T) intersects with edge(A,B)

87) Algorithm for computing the point X where of ray(V,T) intersects with edge(A,B)

88) Algorithm for computing the first intersection of a ray with a polygon

89) Formula for the reflected direction, reflect(V,N), of a ray arriving with direction V on a surface with normal N

6 3D geometry

6.1 Cross-product

90) Explain the properties of the cross-product U×V between two vectors in 3D

91) When is U×V=0?

92) How to compute the normal to triangle(A,B,C) whose length is proportional to the area of the triangle

93) How to compute the unit normal to triangle(A,B,C)?

94) Let s(A,B,C,D) return (AB×AC)•AD>0. Explain why it may be used to test whether triangle(A,B,C) appears clockwise to a viewer at D?

95) Explain how s(A,B,C,D) may be used to test whether point P lies inside tetrahedron (A,B,C,D).

6.2 Distances and intersections in 3D

96) Distance between a point P and line(A,B)

97) Point Q on line(A,B) that is the closest to P

98) Test whether point Q, above, falls inside edge(A,B). Write the test in terms of P, A, and B, not using Q.

99) Distance between a point P and line(A,B)

100) Test whether point P lies inside infinite cylinder, cyl(C,D,r) with radius r and axis through points C and D

101) Closest projection Q of point P onto cyl(C,D,r)

102) Test whether edge(A,B) intersects cyl(C,D,r)

103) Test whether triangle(P,Q.R) intersects cyl(C,D,r)

104) Consider a point P inside triangle(A,B,C) and a direction T that is tangent to triangle(A,B,C). Compute point Q at which ray(P,T) exits triangle(A,B,C).

7 Triangle meshes

7.1 Topological properties

105) Explain the conditions that must be satisfied for a triangle mesh to be water-tight manifold (without border)?

106) Provide a formal definition of when a manifold mesh is connected.

107) How many handles does a connected manifold mesh of T triangles and V vertices have?

108) Assume that two connected manifold meshes U and W do not interfere, how to test whether one is surrounding the other?

109) A shell is a connected manifold triangle mesh. A part is a connected region bounded by one or more shells. You are given 3 shells. How would you test how many parts they bound.

7.2 Orientation

110) How is a triangle orientation in a triangle mesh encoded?

111) Explain the convention for orienting the triangles of the shells that bound a part.

7.3 Corner table construction and operators

112) Explain what a corner is and the Corner Table data structure (what each table contains)

113) Provide the semantics and code for the corner operators: n(c), p(c) o(c), v(c), g(c), t(c), l(c), r(c), b(c), s(c)

114) Assume that the mesh is a shell of n vertices. How many entries are there in the O and V tables?

115) Provide a simple algorithm for computing O from V assuming that V stores the triangles with their proper orientation

116) Provide a simple algorithm for computing O from V not assuming that V stores the triangles with their proper orientation

7.4 Mesh processing and rendering algorithms

117) Provide an algorithm (processing code) or computing the valence of each vertex

118) Provide an algorithm for estimating the normal N[v] of each vertex v

119) Provide an algorithm for smoothing a triangle mesh

120) Provide an algorithm to test whether the edge common to t(c) and t(o(c)) is convex

121) Provide an algorithm to flipping the edge common to t(c) and t(o(c)) and restoring the O and V tables

122) Provide an algorithm for computing the number of shells in a manifold mesh

123) Provide an algorithm for counting the number of border loops (around holes) in a mesh (that is a manifold with border)

124) Provide an algorithm that will test whether a manifold mesh is connected

125) Explain how to test which triangles of a mesh intersect (are completely or partly inside) a ball of center C and radius r.

126) Provide an algorithm to draw the intersection of a triangle mesh with the plane z=h.

7.5 Simplification

127) When is it beneficial to use a simplified model for rendering?

128) How many edge-collapse operations are necessary to reduce the triangle count of a mesh from 10000 to 2000?

129) Provide the code for collapsing the edge common to t(c) and t(o(c)) and restoring the O and V tables

130) Discuss strategies for deciding which edge to collapse next during an iterative mesh simplification process. In particular, explain why picking the edge whose collapse would result in the smallest volume change is not necessarily safe. Provide a simple, even through not the best geometric test fo estimating the error resulting form an edge collapse.

8 Light

8.1 Color and perception

131) How do humans perceive color?

132) Why do we sometimes represent color using RGB components?

133) How does our visual resolution and ability to detect color and movement vary with the position of the object with respect to our gaze direction?

134) Why would two triangles of identical color in an image appear to a human observer to have different intensities? Give several possible reasons.

8.2 Point light source

135) How does the intensity of light emitted by a light source L and received through a small unit receptor surface at point P vary with the distance ||LP||?

136) How does it vary with the angle between PL and the normal N to the receptor surface?

8.3 Surface reflections

137) What is a Lambertian surface? Give one example of such a surface.

138) How is the intensity reflected towards viewpoint V by a small unit region around a point P of planar Lambertian surface affected by the distance d to a point light source at point L?

139) How is the intensity reflected towards viewpoint V by a small unit region around a point P of planar Lambertian surface affected the orientation of the surface normal N when it is illuminated by a point light source at point L?

140) What is the formula for the intensity reflected towards viewpoint V by a small unit region around a point P of planar Lambertian surface with normal N illuminated by a point light source at point L? Is it varying as a function of the distance ||PV||? Is it a function of the distance ||PL||? Is it a function of the angle between N and PV? Is it a function of the angle between N and PL? Justify each answer.

141) When can one assume that the intensity of light reflected by a Lambertian surface is constant along an unobstructed ray?

142) What is a specular surface? What is the most extreme example of a specular surface?

143) What is the formula commonly used for approximating the intensity of the light emitted from light source point L and reflected towards a viewer V by a unit region of a specular surface around a point P with normal N.

144) Why is the above formula wrong when the specular surface (wall) is lit by a candle in the room? Give a convincing example and then explain how to fix the formula so that it is correct. Explain when the common formula is a safe approximation of the correct one.

145) What is the light reflection formula for an arbitrary surface? Discuss what each term means.

146) You are at point V in a room with a mirror wall (plane through point Q and normal N) and a candle at L. Provide the formula for computing the point P on the mirror wall where you see the candle. Provide the formula for the intensity perceived.

147) Explain the difference between flat, Gouraud and Phong shading and give examples of the benefits of using Phong shading.

148) If you use Gouraud shading to render a scene, the image will be incorrect for several reasons. Give two three.

8.4 Ray tracing

149) You are given a 3D model of a screen and viewpoint in the scene. Explain how you would generate all primary rays from the viewpoint through the center of each pixel.

150) Provide the high-level algorithm for ray-casting and discuss its computational and storage cost.

151) How to test whether ray from P in direction T intersects triangle(A,B,C).

152) Compute the point X where a ray from P in direction T intersects triangle(A,B,C), assuming the intersection exists.

153) Test whether ray(V,T) hits triangle(A,B,C)

154) How would you compute the first point that a given ray(V,P) hits in a scene?

155) What secondary rays would you shoot from there and why? How would you use these secondary rays?

156) The viewer is at V and is looking at a specular triangle(A,B,C). The light point is at L. Explain how you would compute the highlight and check whether it is inside the triangle.

157) The scene has only 2 triangles: T1=triangle(A,B,C) and T2=triangle(D,E,F) that are made of a specular surface. You will render them through ray tracing. Explain which primary rays you will need to shoot and how you could quickly establish these. Explain how for each one of these, you would compute the point P where the primary ray first intersects one of these triangles. Then explain precisely which secondary rays you will shoot from P and how you will process them to render shadows. Finally, explain what you could do to compute highlights (with one or two bounces).

8.5 Rasterization, shaders, and GPU architecture

158) What is a z-buffer? What information does it contain? Where is it stored? How much storage does it use? What is it used for? What would one have to do if it were not available?

159) Which pixels are modified when rasterizing a line? Provide a clear an unambiguous formula.

160) Explain the input and output of a triangle rasterizer and precisely what its job is.

161) Which pixels are modified when rasterizing a triangle?

162) List the main stages of the rendering pipeline and for each stage give a brief outline of its input, output, and job.

163) Draft the algorithm for rasterization, contrast it with the ray-casting algorithm, discuss its advantages and drawbacks.

164) Explain the per-pixel and per-vertex costs, and for each give an example when it would be the bottleneck in the graphics pipeline.

165) Why are triangles clipped? Which stage of the pipeline is responsible for that?

8.6 Perspective

166) Given a point P=(x,y,z) in screen coordinates (origin is at the center of the screen) and a viewpoint V=(0,0,-d), provide the formula for point P'=(x',y',z') that is the image of P by the perspective transform.

167) Which pixel will P' be rendered at? Why is this correct?

168) What is the perspective projection of P?

169) Why can't we use the perspective projection for rasterization? Why do we need the perspective transform?

170) Why can we not set z'=z in the perspective transform?

171) How do you draw a correct perspective of a block given its front corner and the 3 vanishing points?

172) Is that perspective drawing correct no matter where you are looking from at the image?

173) Given the pixel (x',y') at which a point P is rendered and the z' value stored in the zbuffer z[x',y'] at that pixel, how do you recover the screen coordinates (x,y,z) of P?

174) What is the geometric construction of the perspective transform P' of a point P? Draw it in 2D.

175) Where are points at infinity mapped under perspective transform?

176) Why do we sometimes use near and far clipping planes?

177) Consider two triangles that are parallel and nearly coincident. One just behind another. The one in front is green, the other one red. Initially, the red triangle is hidden. You slowly move them towards or away from the viewer. You notice that at some point, the red triangle appears as if in front of the green one. Explain what have caused it? Did this happen when you moved the triangles away from the viewer or closer? What can you do to reduce the probability of this happening?

8.7 Texture mapping

178) Processing code to shade a polygon with vertices {Pi}

179) Processing code to shade and texture map a polygon with vertices {Pi} assuming that the texture is already active. Explain what the texture coordinates mean.

180) You have a square image I in which there is a horizontally aligned profile view of a small car. The user has traced a loop (lasso) on I around that car. The lasso is represented as the polyloop {Pi}. Explain how you would produce an animation showing that car sliding on the smooth 2D curve that may represent the ridge of a hill (we want to animate a side view of a small car driving up and down the hill along that ridge). We will represent the hill as a cubic Bezier curve. Explain how you would generate consecutive points on it for the consecutive frames of your animation and how you would compute the tangent to the curve at the current point. Then, explain how you would display the cutout image of the car so that it appears to drive on the curve while remaining aligned tangentially with the curve. First explain your solution at a high level, then provide the as much details (Processing code) as you can.

8.8 Shadows, mirrors, occlusion

181) Explain how to produce the image of a scene in a room where one of the walls is a mirror and where we see floor shadows cast by a source light infinitely high.

182) List 3 techniques for producing images of a 3D scene with shadows and briefly comment on their advantages and drawbacks.

183) You want to render an image of a 3D model in a square room where two adjacent walls are mirrors. Explain how you can do this using the graphics pipeline.

184) You want to render a park where the same tree is to be rendered at different locations. To render an instance of the tree at point P you only have to call showTree(P). However, between the camera and the park there is a large and tall box lying on the ground that occludes some of the trees. You want to quickly identify which trees are occluded. Explain how you would do this (what you would pre-compute, how you would use it).

8.9 Shaders and effects

185) What is a fragment? What stage of the graphics pipeline produces them?

186) Explain the default role of vertex and fragment shaders.

187) Which shader would you modify to implement Phong shading and what would it do.

188) Explain precisely how environmental mapping works and what it is useful for.

189) List 3 effects that increase realism and may be rendered on the GPU. For each briefly explain why they are important and how they may be implemented.

190) What are stencil planes and what can they be used for.

8.10 NPR

191) What does NPR stand for?

192) Explain how to compute the silhouettes on a triangle mesh

193) Justify that these silhouettes for closed loops

194) Give examples of how images drawn by artists may differ from those produced by standard rendering, for each style effect give a name, explain what it is, and point out why this is valuable

195) Describe a simple scene where flat shading will produce a very confusing image. Explain how NPR may be used to make the image easier to understand.

8.11 GENERAL

196) Define the field of computer graphics and give examples of remaining challenges for software developers

197) Inventing new computer graphics techniques may require logic, math, intuition, and experimentation. Give an example of each from your own experience (in class projects, homework, jobs).

198) List 3 of the major applications of computer graphics that are not entertainment related and explain what 3D graphics and animations are used for in these applications and why they are important.

199) List 3 of the most significant problems, which must be solved for 3D graphics to realize its full potential in society. For each, briefly explain what the problem is, why it is hard, and what would be the fundamental benefit of solving it.

200) List 3 principles used by the feature animation artists for creating pleasing animations and explain which techniques discussed in class would help you support such principles.