CSC 321 Final Exam (2015)

Instructions:

1. This is a take-home, open-book, and open-notes exam. You may use calculators and computers.
2. Do not discuss any exam-related question with anyone except the instructor (email: rsowell@cornellcollege.edu).
3. The time limit is 5 hours, which can be broken into maximally 5 blocks completed at different times.
4. The completed exam with signed cover sheet (this page) is due by **Wednesday, April 8th at 12:00 PM**.
5. Please list any books, websites, or any other sources that you used besides the lecture notes for this course.

By signing below, I indicate that I have read, understood and followed the instructions above.

Name: ______________________________________.

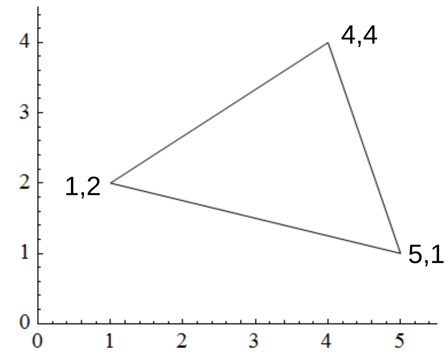
Signature: ____________________________________.

Date : ____.

Time used: ________hours_______minutes

1. Scan conversion and transformations **[10 pts]**

    1. Write down the x,y coordinates of the pixels that are turned on when scan converting the three edges of the triangle in the picture (any choice of rounding at 0.5 is fine). **[2 pts]**



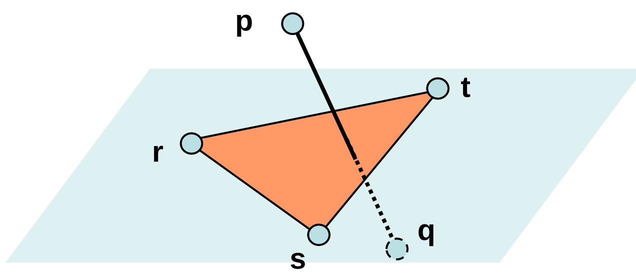
    2. What is the transformation matrix (write as products of **T**,**S**,**R**) that rotate this triangle 90 degrees counter-clockwise around its lower-left corner (2,2) **[1 pts]**? Calculate the final 3 by 3 transformation matrix **[2 pts]**.

    3. Using this matrix, compute the new locations for the three vertices of the triangle after transformation (using homogeneous coordinates). Show your matrix/vector multiplications. **[3 pts]**

    4. Write down the x,y coordinates of the pixels that are turned on when scan converting the three edges of the rotated triangle. **[2 pts]**

2. Points and vectors: compute the intersection between a line segment and a triangle in 3D. Your answers should be presented using point and vector operations (e.g., addition, subtraction, magnitude, dot and cross products) and no x,y,z coordinates. **[6 + 2 pts]**

Consider a line segment with end points **p**, **q**, and a triangle with vertices **r**,**s**,**t**, see below:



1. Without computing the intersection point, <u>write down</u> the test that checks whether the line segment intersects with the *plane* on which the triangle lies (Hint: think about what side of the plane that **p**, **q** needs to be on). **[3 pts]**

2. Assuming the line segment passes your test, <u>compute</u> the location of that intersection point. **[3 pts]**

3. **[Extra credit]** How would you test if the intersection point computed above lies *inside* the triangle? **[2 pts]**

3. **Camera projection**. Use look vector {-1,0,0}, up vector {-1,1,1}, height and width angle 90 degrees, near clipping plane at 0.001. **[9 + 2 pts]**

1. Compute camera coordinate system axes u,v,n. **[2 pts]**

2. Let the view point be at {2,0,0} and the far clipping plane be at distance 5**.** Write the matrices involved in the world-to-camera transformation, and compute their product 4by4 matrix. **[4 pts]**

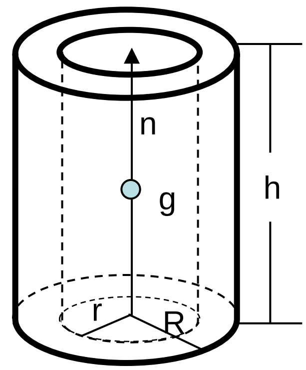
3. Place a unit-radius sphere at the origin. Compute the location of the sphere center and the radius of the sphere after the world-to-camera transformation. **[3 pts]**

4. (**Extra credit**) Fixing the rest of the camera parameters, where should the *far clipping plane* be so that the sphere center after the world-to-camera transform is at (0,0,-½)? **[1 pt]**

5. (**Extra credit**) Fixing the rest of the camera parameters, where should the *viewpoint* be so that the sphere center after the world-to-camera transform is at (0,0,-½)? **[1 pt]**

4. Ray Intersection **[10 + 2 pts]**. Consider the cylindrical shell on the right. The shell is made by cutting out a smaller cylinder from a bigger cylinder. The shell is centered at **g** with axis in the direction of a unit vector **n**. The height of the shell is **h** and the radii of the inner and outer cylinder are respectively **r** and **R** (**R**>**r**).

1. <u>Write down</u> the implicit equations, one for each part of the surface of the cylindrical shell (there are 4 parts). **[4 pts]**

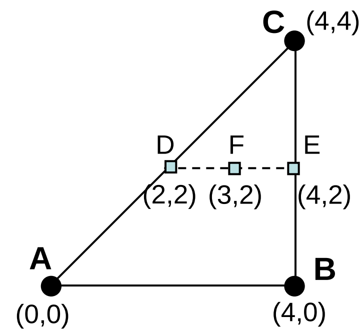


2. Using the sign of these equations, <u>write down</u> the test that checks if a point **p** lies within the shell (not within the inner cylinder, but in the solid shell between the two cylinders) Your test should read like "p lies inside if f1[p]<0 and f2[p]>0 …"). **[2 pts]**

3. Given a ray defined by a point **q** and a direction **v**. <u>Write down</u> the tests that check if this ray intersects the shell, and if so, which part of the shell surface it intersects. Your tests should read like "the ray intersects the outer cylindrical side if …". You only need to present the equations and inequalities without having to solve them). **[4 pts]**

4. **[Extra credit]** Make sure that your tests in question 3 are minimal, that is, if any checks are taken off your test, the test will become incorrect (if your tests are already minimal in question 3, you get this credit automatically) **[2 pts]**

5. Polygon Shading **[10 + 2 pts]**. A triangle has already been projected into the screen coordinate space, as shown on the right. Assuming that the *un-normalized* normals associated with each un-projected vertex A, B, and C are (in world coordinates) (-1,0,1), (1,0,1), (0,1,1), and the only light source in the 3D environment is a directional light in the direction (in world coordinates) (0,0,-1) with RGB color (1,1,1). Use diffuse coefficient (1,1,1) for the following problems, and show your calculations as much as possible for partial credits.



1. <u>Compute</u> the normalized normal vector at each triangle vertex. **[1 pt]**

2. Use Gouraud shading, <u>compute</u> the *red* component of the *diffuse* color at points D,E,F. **[4 pts]**

3. Use Phong shading, <u>compute</u> the *red* component of the *diffuse* color at points D,E,F. **[5 pts]**

4. **[Extra Credit]** In the following graph, <u>plot</u> (roughly) how the red color changes along the line DE under Gouraud shading and Phong shading. In the graph, the horizontal axis is the line from D to E, and vertical axis is the red component of the diffuse color (between 0 and 1) at a point on that line. (Hint: you already know 3 points on each graph from previous two problems. The question is what kind of curve connects those three dots) **[2 pts]**